

## КОНСЕНСУС ЕКСПЕРТІВ ЯК ЗАСІБ ДЛЯ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ МОДЕЛЕЙ ЗНАНЬ В ІНТЕЛЕКТУАЛЬНИЙ СИСТЕМІ

© Буров Є., Крамаренко М., 2011

**Розроблено метод та алгоритм для верифікації моделей знань інтелектуальної системи, який ґрунтується на консенсусі групи експертів. Показано місце моделі консенсусної верифікації серед задач верифікації моделей, наведено схеми відповідних моделей. Визначено вимоги до організаційної підтримки та супроводу процесу верифікації моделей, а також описано механізми підтримки виконання моделей.**

**Ключові слова:** верифікація моделі знань, консенсус експертів/

**Described method and algorithm for knowledge model verification based on consensus of expert group members. Shown the place of developed method and corresponding model among other verification models, developed schemata of those models. Defined requirements for organizational support of model verification process, described details of model execution implementation.**

**Keywords:** knowledge model verification, expert consensus.

### Постановка проблеми у загальному вигляді

Однією з найважливіших проблем галузі програмного забезпечення є високий рівень складності програмних систем і пов'язані з нею проблеми складності та високої вартості адміністрування, розроблення та модифікації, значний рівень дефектів у таких системах. За оцінками національного інституту стандартів та технологій (NIST), щорічний збиток від дефектів програмного забезпечення для економіки США оцінюється у 59,6 млрд. доларів, при цьому тільки третину цих дефектів можна усунути шляхом застосування традиційних методів контролю якості програмного забезпечення, таких як тестування [1]. Традиційний підхід до створення програмних систем на базі фіксованого набору вимог призводить до створення негнучких систем, які дорогі у супроводі та погано функціонують в умовах зміни бізнес-середовища [2].

Одним із шляхів вирішення зазначеної проблеми є використання моделей для автоматизації побудови та функціонування програмних систем – MDA/MDD – Model driven architecture/Model driven design [3]. Найвідомішим підходом до побудови програмних систем на основі моделей є xUML [4]. За цим підходом спочатку будують формальну специфікацію системи у вигляді набору моделей, що відображають різні сторони її функціонування, такі як інтерфейс користувача, програмну логіку, а потім компілюють ці моделі у виконувальний код. Недоліком такого підходу є висока складність як самих специфікацій, так і процесу їх компіляції в єдиний продукт, яка нерідко порівнянна зі складністю традиційного процесу розроблення [5,6]. У роботах [7, 8] було запропоновано спосіб побудови програмних систем на основі простих інтерпретованих моделей, які будують з використанням онтології предметної області. На відміну від моделей xUML, такі моделі можуть створюватися та модифікуватися фахівцями конкретних предметних областей, а не тільки програмістами, вони є простими, дають змогу повторно використовувати знання. Складна задача розв'язується в процесі взаємодії багатьох моделей, кожна з яких розв'язує певну достатньо просту підзадачу.

Моделі для розв'язання задач створює фахівець предметної області, який повинен не тільки відобразити у моделі усі важливі для поставленої задачі сутності та відношення предметної області, але й визначити обмеження на застосування моделей, необхідні умови щодо наявності вхідних

даних, визначити коло задач для яких метод реалізований моделлю, є релевантним, та відобразити умови релевантності через властивості сутностей предметної області. Безпомилковість, відсутність протиріч та повнота моделей є ключовими факторами, який визначають якість рішень, які виробляються з використанням моделей.

Розроблення моделей знань передбачає наявність високого рівня компетентності розробника і є важко формалізованою задачею. З іншого боку, помилка у моделі призведе до помилок при розв'язанні усіх задач, в яких застосовується ця модель. Тому важливою стадією розроблення моделей є їх верифікація – як початкова, що проводиться автором моделі, так і додаткова, яку проводить комісія з експертів предметної області.

Уведення надлишковості, тобто використання групи експертів для додаткової верифікації моделі є засобом підвищення достовірності відображених у моделі знань за рахунок деякого збільшення вартості розробки. Певною проблемою є те, що експерти самі можуть помилятися і навіть звичайне голосування експертів, тобто виявлення позиції більшості експертів, також не гарантує від помилки.

Достовірність рішення експертів можна підвищити, якщо увести додатковий етап у прийнятті рішення, який полягає у порівнянні експертних висновків з метою виявлення явних помилок, та вилучення з процесу прийняття рішення експертних висновків, в яких виявлено помилки.

### **Аналіз останніх досліджень та публікацій**

Запропонований підхід ґрунтується на уведенні фази експертної верифікації розроблюваних моделей знань. Для верифікації моделі організують групу експертів (рецензентів), кожен з яких перевіряє модель та дає обґрунтовану оцінку, зазначаючи переваги та недоліки моделі. Вважатимемо, що серед експертів немає конфлікту інтересів.

Очевидно, що рецензент може допустити помилки під час верифікації моделі. Рецензент, який допустив помилку, може бути вилучений зі складу експертної групи з подальшим зниженням його рейтингу, який використовується при створенні нових експертних груп.

Завданням експертної групи є досягнення консенсусу щодо моделі, яку оцінюють. Під консенсусом вважатимемо отримання узгодженої оцінки, яка ґрунтується тільки на оцінках, що не містять помилок.

Варто зазначити, що задача отримання консенсусу розв'язувалася у межах підходу до побудови відмовостійких обчислювальних систем. Тут можна виділити два підходи [9]. Перший підхід розробляється в межах побудови моделей діагностики відмов на системному рівні (system-level fault diagnosis [10]) та виходить з класичної роботи [11]. Другий підхід розробляється в межах аналізу можливості досягнення т.зв. «візантійської угоди» та виходить з класичних робіт [12,13].

Обидва підходи мають подібну мету: визначення правильного результату в умовах відмов за рахунок уведення надлишковості. Перший підхід ґрунтується на виявленні дефектних компонент системи за рахунок організації взаємних перевірок. При цьому вважають, що кожен працездатний модуль системи правильно визначає відмову модуля, який він перевіряє, а кожний дефектний модуль видає непередбачуваний результат про стан модуля, що перевіряється. Отримані результати взаємних перевірок (синдром) аналізуються глобальним арбітром (global observer або centralized arbiter [9]).

Для визначення умов централізованого дешифрування синдрому вводиться параметр  $t$  – максимальна кількість дефектних модулів [14]. У [14] показано, що для дешифрування синдрому повинна виконуватися умова  $n \geq 2t + 1$ , де  $n$  – кількість модулів системи.

Другий підхід ґрунтується на децентралізованому прийнятті рішення про правильність результатів обчислень. Тут вводять комунікаційну та протокольну надлишковість, яка і дає змогу досягти «візантійської угоди» за умови  $n \geq 3t + 1$ , де  $t$  – кількість компонентів системи, які спотворюють результати обчислень [12].

Зазначимо, що припущення про наявність централізованого дешифрування синдрому цілком припустиме для випадку організації верифікації моделі знань інтелектуальної інформаційної системи, оскільки таке завдання може бути інтегроване у саму модель підтримки прийняття рішення шляхом консенсусу.

## Цілі статті

Метою цієї статті є розроблення методу верифікації моделей знань інтелектуальної системи, що ґрунтується на досягненні консенсусу групи експертів та визначення способів використання цього методу як окремої моделі знань – складової частини інтелектуальної системи.

### Метод та алгоритм отримання консенсусного рішення у процесі верифікації моделі

Нехай кожен експерт (рецензент) оцінює верифіковану модель знань з використанням цілочислової додатної шкали вигляду  $S = \{i_{\min}, \dots, i_{\max}\}$ . Тоді кожній оцінці  $j$  відповідає інтегральна рецензія (позитивна або негативна), яку отримаємо, використовуючи вираз  $r_j = (i_{\max} - i_{\min})/2 \cdot j$ . Результат цього виразу переводить оцінку експерта в інтегральну рецензію (або просто рецензію), яку подамо булевою змінною зі значенням 1 для від'ємних  $r_j$ , та 0 в іншому випадку.

Множину рецензій подамо зваженим графом  $G(V,E)$  без циклів, множина вершин якого подає множину рецензій, а кожне ребро  $e_{ij} \in E$  відповідає операції порівняння рецензій  $v_i, v_j$ . Вага такого ребра (булева змінна  $a_{ij}$ ) подає результат порівняння рецензій. Будемо називати такий граф графом порівняння рецензій. Будемо також називати працездатною вершину цього графу, якщо відповідна рецензія не є помилковою, і непрацездатною в іншому випадку.

Матрицю ваг ребер графу подають синдромом  $\delta = \{a_{ij}; (i,j) \in E\}$ . Можливі такі результати виконання операції порівняння рецензій.

Нехай верифікована модель правильна і результати рецензування збіглися. Збіг результатів рецензування можливий, тоді і тільки тоді, коли обидва рецензенти визначили модель знань як правильну (і не допустили помилки), або обидва рецензенти вважають модель хибною (і обидва помиляються).

Аналогічно, нехай верифікована модель містить помилки, а рецензії збіглися. Збіг рецензій можливий, якщо і тільки якщо обидва рецензенти визначили модель як правильну (обидва помилилися) або обидва визначили модель як хибну (тобто прийняли правильне рішення).

З цього випливає, що операція порівняння рецензій визначає можливі ваги дуг за такими правилами:

$a_{ij} = a_{ji} = 0$ , тоді і тільки тоді, коли рецензії збіглися (тобто рецензенти  $u_i, u_j$  прийняли однакове рішення – помилкове або правильне);

$a_{ij} = a_{ji} = 1$ , в інших випадках (якщо хоч би один з рецензентів  $u_i, u_j$  помилився).

Можливі значення ваг дуг зручно визначати такою четвіркою [14, 15]:

$a_{gg}$  – рішення, яке приймає працездатна вершина про стан іншої працездатної вершини, яка перевіряється;

$a_{gb}$  – рішення, яке приймає працездатна вершина про стан іншої, непрацездатної вершини;

$a_{bg}$  – рішення, яке приймає непрацездатна вершина про стан іншої, працездатної вершини;

$a_{bb}$  – рішення, яке приймає непрацездатна вершина про стан іншої, також непрацездатної вершини.

Очевидно, що  $a_{gg} = 0$ ;  $a_{gb} = 1$ . Припускаємо, що  $a_{bg}, a_{bb} \in \{0,1,-\}$ , де символ “-” позначає непередбачуваний результат 0 або 1.

Отже, операція порівняння рецензій визначає перевірочне відношення четвіркою 0110.

**Теорема 1** [15]. Синдром завжди симетричний відносно головної діагоналі, тоді і тільки тоді, якщо  $a_{gb} = a_{bg}$ ;  $a_{bg}, a_{bb} \in \{0,1\}$  і для кожної дуги  $e_{ij} \in E$  існує дуга  $e_{ji} \in E$ .

**Наслідок.** Граф порівняння рецензій завжди є неорієнтованим.

Зазначимо, що модель діагностики відмов на системному рівні, визначена четвіркою 0110, досліджена у роботах [14, 15]. В них показано, що для дешифрування синдрому необхідно обмежити кількість непрацездатних вершин  $t$  формулою  $n \geq 2t+1$ , де  $n = |V|$ .

З цього випливає:

**Теорема 2.** Для досягнення консенсусу необхідно, щоби виконувалося  $n \geq 2t+1$ . Якщо  $n \geq 2t+1$ , то консенсус існує.

**Необхідність.** Розглянемо повнозв'язний граф, в якому  $n=2t$ . Тоді, за визначенням моделі, у такому графі можна утворити два різні цикли довжини  $t$  з нульовими вагами, один з яких містить

тільки працездатні, а інший – тільки непрацездатні вершини. При цьому ребра, що з'єднують вершини, які належать різним циклам, матимуть одиничні ваги. Тому такий синдром неможливо розшифрувати.

**Існування.** Розглянемо повнозв'язний граф, в якому  $n \geq 2t + 1$ . Тоді, за визначенням моделі, довільний цикл з нульовими вагами ребер довжиною, більшою за  $t$ , вказуватиме на працездатність вершин, які належать йому. Ребра з одиничними вагами, що відходять від цього циклу, дадуть змогу виявити непрацездатні вершини.

На рис. 1,2 наведено приклади 2-діагностованого та недіагностованого графів.

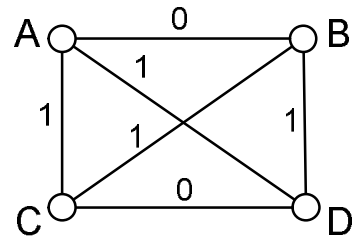
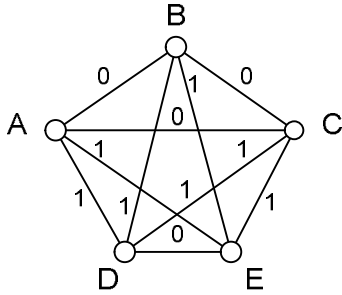


Рис. 1. Приклад 2-діагностованого графу. На рисунку вершини A,B,C – працездатні, а D,E – непрацездатні

Рис. 2. Приклад недіагностованого графу. Невідомо, які вершини непрацездатні (A, B, або C,D)

**Теорема 3.** Довільний простий шлях у графі з ребрами нульової ваги і довжиною більшою за  $t$ , складається з працездатних вершин.

**Доведення.** За визначенням моделі, довільний простий шлях з нульовими вагами вершин повинен складатися з тільки працездатних або з тільки непрацездатних вершин. Кількість непрацездатних вершин обмежено числом  $t$ . Тому довільний шлях з нульовими вагами ребер довжини, більшої за  $t$ , можна утворити тільки з працездатних вершин.

**Теорема 4.** У повнозв'язному графі, якщо  $t \leq (n-1)/2$ , то довільний рядок синдрому однозначно дешифрує синдром.

**Доведення.** Нехай  $t=0$ . Тоді усі елементи кожного рядка синдрому матимуть нульову вагу.

Нехай  $1 < t \leq (n-1)/2$ . Тоді, з визначення моделі та повноти графу випливає, що на основі синдрому можна побудувати два повні підграфи вихідного графу, причому кожен такий підграф матиме ребра з нульовою вагою, а усі інші ребра вихідного графу матимуть одиничну вагу. При цьому менший підграф, що міститиме непрацездатні вершини, матиме  $t \leq (n-1)/2$  вершин, а більший підграф – складатиметься з решти  $n - t$  вершин (рис 3). Отже, кожен рядок синдрому міститиме тільки нульові елементи.

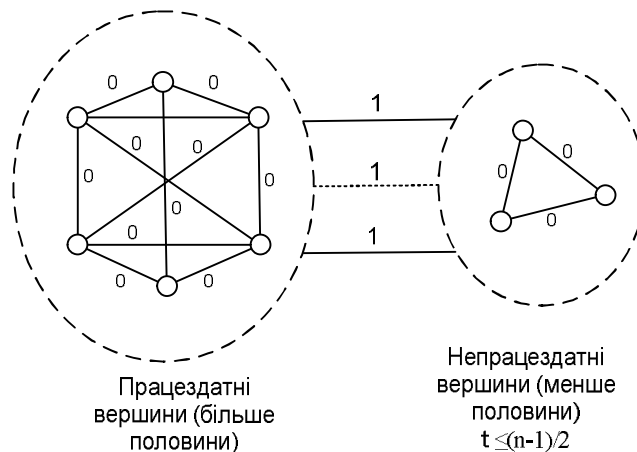


Рис. 3. Ілюстрація до теореми 4

Довільна працездатна вершина  $v_j$  буде однозначно ідентифікуватися нульовими елементами в рядку  $j$  синдрому, якщо і тільки якщо їх кількість буде більшою за половину кількості вершин вихідного графу. У цьому випадку кожен одиничний елемент рядка буде однозначно ідентифікувати непрацездатну вершину.

Довільна непрацездатна вершина  $v_j$  буде однозначно ідентифікуватися нульовими елементами у рядку  $j$  синдрому, якщо і тільки якщо їх кількість буде менша за половину вершин вихідного графу. У цьому випадку кожен одиничний елемент рядка однозначно визначатиме працездатну вершину.

Нехай тепер  $t=1$ . Тоді менший підграф стягнеться в одну вершину, усі ребра якої матимуть одиничну вагу, тобто буде порушено обмеження, визначене теоремою 2. У цьому випадку усі ребра вершин більшого підграфу, інциденті непрацездатній вершині, матимуть одиничну вагу, а решта інцидентних їм ребер – нульову вагу.

З цього випливає:

Алгоритм виявлення непрацездатних вершин. Для визначення непрацездатних вершин достатньо знайти усі нульові елементи довільного рядка синдрому. Приймають рішення за такими правилами:

- Якщо рядок містить тільки нульові елементи, то непрацездатних вершин немає.
- Якщо рядок містить тільки одиничні елементи, то він відповідає єдиній непрацездатній вершині.
- Якщо їх кількість менша за половину кількості вершин, то рядок відповідає непрацездатній вершині графу. Тоді усі одиничні елементи вкажуть на усі працездатні вершини графу, а нульові – на усі непрацездатні вершини.
- Якщо їх кількість більша за половину кількості вершин, то рядок відповідає працездатній вершині графу. Тоді усі одиничні елементи вкажуть на непрацездатні вершини, а нульові – на працездатні вершини графу.

Очевидно, що складність алгоритму можна оцінити як  $O(n)$ .

Отже, для досягнення консенсусу рецензентів достатньо:

1. Виконати  $n*(n-1)/2$  операцій порівняння рецензій.
2. Побудувати синдром, що визначає ваги ребер графу рецензій.
3. Виконати алгоритм виявлення непрацездатних вершин.
4. Визначити рецензентів, що помилилися, та вилучити їхні оцінки з подальшого розгляду.
5. Зберегти список рецензентів, що помилилися, для формування груп експертів у майбутньому.
6. Прийняти консенсусне рішення щодо верифікованої моделі знань.
7. Зберегти сумарну оцінку верифікованої моделі знань для накопичення сумарного рейтингу автора (фахівця предметної області).

### **Побудова моделей знань для верифікації концептуальних моделей підтримки прийняття рішень**

Розв'язання задачі верифікації моделей є повторюваною бізнес-процедурою в інтелектуальній системі, що ґрунтується на моделях. Для її проведення важливо дотримуватися єдиних підходів, рекомендацій та правил, що забезпечить єдині вимоги до комплексу моделей, що використовуються. Такі підходи відображають технічну політику організації у питанні верифікації моделей та є складовою частиною корпоративних стандартів цієї організації. Завдання верифікації моделей є складним та важкоформалізованим і тому вирішується шляхом експертного оцінювання.

Водночас система оцінювання моделей повинна бути достатньо гнучкою та надавати можливість застосування різних методів оцінювання. Застосування того чи іншого методу залежить від змісту та призначення моделі, наявних ресурсів. Для вироблення такого єдиного підходу необхідно розв'язати такі задачі:

- забезпечити формалізацію комплексу моделей знань і так створити можливість їх автоматизованого виконання;

- забезпечити можливість використання різних методів оцінювання моделей і для цього створити таксономію (онтологію) типів моделей, об'єднаних спільним завданням – верифікацією моделей шляхом їх оцінювання;
- надати менеджеру, що керує процесом впровадження та використання моделей, засоби для контролю та керування процесом оцінювання;
- реалізувати можливість повторного використання моделей та знань, поданих моделями, уникнути дублювання знань у моделях.

### Ієрархія методів та моделей верифікації

Моделі оцінювання утворюють ієрархію, в корені якої знаходиться модель оцінювання, подана у найзагальнішому, абстрактному вигляді – загальна, базова модель. Ця модель використовується у системі моделювання для формування запиту на розв'язання задачі та подання найзагальніших параметрів цього запиту. На основі цих параметрів компонент системи моделювання – брокер взаємодії моделей – обирає адекватний до запиту метод розв'язання поставленої задачі, поданий тією чи іншою моделлю.

У роботі [8] подано головні принципи побудови виконувальних концептуальних моделей та організації їх взаємодії. Модель Md складається зі схеми ScMd та реалізації RIMd:

$$Md = (ScMd, RIMd)$$

Схема моделі безпосередньо видима користувачу та проектувальнику моделі – фахівцю визначеної предметної області. Проектувальник може змінювати схему, створюючи похідні моделі. Реалізація моделі створюється фахівцем- програмістом та забезпечує виконання моделі. Загальна модель виступає тільки як схема, тобто вона не має реалізації. Кожна модель таким чином розглядається як складова частина певної ієрархічної системи моделей, що реалізують різні методи розв'язання схожих за постановкою задач, доповнюючи або замінюючи одна одну залежно від особливостей конкретної задачі. Розглянемо спосіб використання моделі консенсусу експертів на прикладі ієрархії моделей, до якої входять загальна модель, модель прийняття рішення голосуванням та модель консенсусу експертів.

### Загальна модель

Загальну модель *Evaluation* подамо як набір таких сутностей та відношень (рис 4):

Сутність *EvaluatedObject*. Визначає оцінюваний предмет. У загальному випадку оцінювати можна довільний предмет, отже

$$Type(EvaluatedObject) = Thing,$$

де *Type()* – це функція, що повертає тип онтології, якому відповідає сутність-аргумент, а *Thing* – кореневий елемент онтології.

Сутність *Evaluator*. Визначає сутність, яка формує оцінку предмета оцінювання.

Відношення *Evaluate*. Визначає саму операцію оцінювання та поєднує суб'єкт та об'єкт оцінювання (*Evaluator* та *EvaluatedObject*). Важливим атрибутом відношення *Evaluate* є сутність *Value*, визначення якої і є результатом виконання моделі.

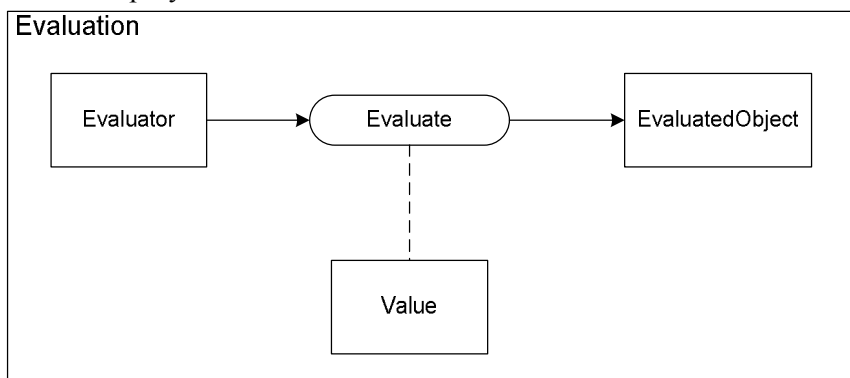


Рис. 4. Схема загальної моделі оцінювання

Як видно з визначення моделі *Evaluation*, вона відповідає широкому класу задач. При цьому задача оцінювання моделі є частковим випадком загальної задачі оцінювання. Для нього визначимо додаткові обмеження на елементи загальної моделі, які передаються при звертанні до неї у процесі активації. Так, предметом оцінювання є модель, тобто  $Type(EvaluatedObject) = Model$ . Крім того, передбачається, що оцінка моделі обирається з дискретної шкали оцінок (*true*, *false*). Отже,

$$Range(Value) = (true, false),$$

де *Range()* – функція, що повертає область значень для атрибутивної сутності *Value*.

Достатньою умовою для активації загальної моделі є визначеність сутностей – ролей *Evaluator* та *EvaluatedObject*. У випадку, якщо в запиті на вирішення задачі оцінювання явно не вказано оцінювача, його може визначити брокер взаємодії моделей на основі наявної інформації про предмет оцінювання та вимоги до оцінки. Умовою успішного виконання моделі є визначеність атрибутивної сутності *Value*.

### Модель консенсусного оцінювання

Модель консенсусного оцінювання має два послідовні етапи виконання (рис 5). На першому етапі проводиться первинне рецензування оцінюваного об'єкта, поданого сутністю *Model*. Передумовою активування моделі та початку першого етапу є визначеність об'єкта оцінювання (моделі) та експертної комісії. В результаті виконання першого етапу формуються впорядкована множина оцінок *OrderedValueSet*. Умовою завершеності першого етапу виконання моделі є завершеність формування його результатів. На рисунку результати виконання кожного етапу подано прямокутником із сірим заповненням. Виконання першого етапу моделі може потребувати довшого часу, адже експертам потрібно мати час на аналіз моделі та створення оцінок. Система моделювання відслідковує завершеність окремих етапів і повідомляє менеджера про їх статус та завершеність.

Другий етап виконується автоматично, без участі експертів. Операція аналізу результатів визначає групу експертів, які не зробили помилок в оцінках та визначає рішення, прийняте консенсусом експертів. У результаті виконання цього етапу формується консенсусне рішення *Value* та оновлюється рейтинг експертів, які брали участь в оцінюванні.

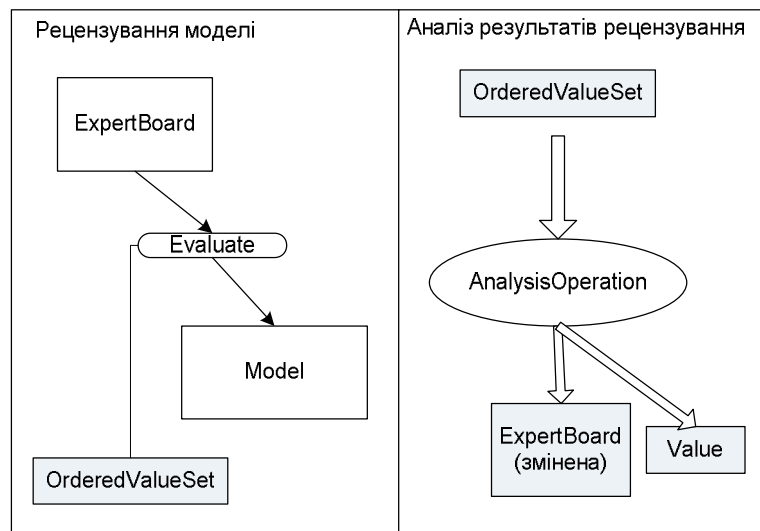


Рис. 5. Схема моделі консенсусного оцінювання

### Механізми виконання та підтримки взаємодії моделей

Закінчена модель подається xml- файлом, який містить секції метаданих, ініціалізації, передумов, посилання на загальну модель, тіла моделі, верифікації та ін. [8]. Опис моделі зберігається у репозиторії моделей, який разом з онтологією та інформаційною базою є складовою частиною бази знань інтелектуальної системи (рис 6). Онтологія забезпечує семантичну інтерпретацію фактів з інформаційної бази та дає змогу сформулювати найзагальніші обмеження щодо

сутностей та відношень, що входять до неї. Моделі формують виключно з використанням сутностей та відношень, поданих в онтології.

Модель виконується інтерпретатором моделей. Для спрощення реалізації інтерпретатора моделей доцільно задавати логіку його роботи у вигляді xml- файлу, який послідовно визначає окремі кроки алгоритму виконання моделі. На кожному кроці відбувається звертання до однієї з типових операцій, які підтримує інтерпретатор, таких як перевірка передумов моделі, отримання даних з інформаційної бази, активація зовнішньої моделі та очікування результатів від неї тощо. Важливою вимогою до інтерпретатора моделей є ведення детального протоколу моделювання, пояснення кожного кроку у термінах, зрозумілих менеджеру-фахівцю предметної області.

Взаємодії моделей досягають за посередництвом окремого компонента системи моделювання – брокера взаємодії моделей. Модель, яка потребує вирішення додаткової, допоміжної задачі, формує свої потреби у вигляді загальної, абстрактної постановки задачі, поданої як загальна модель. Загальна модель використовується для розв’язання широкого кола подібних задач, що мають спільну постановку. Так, загальну модель *Evaluation* можна використати не тільки для оцінювання моделей знань, але й для оцінювання рівня підготовки студентів, якості послуг у магазині та ін.

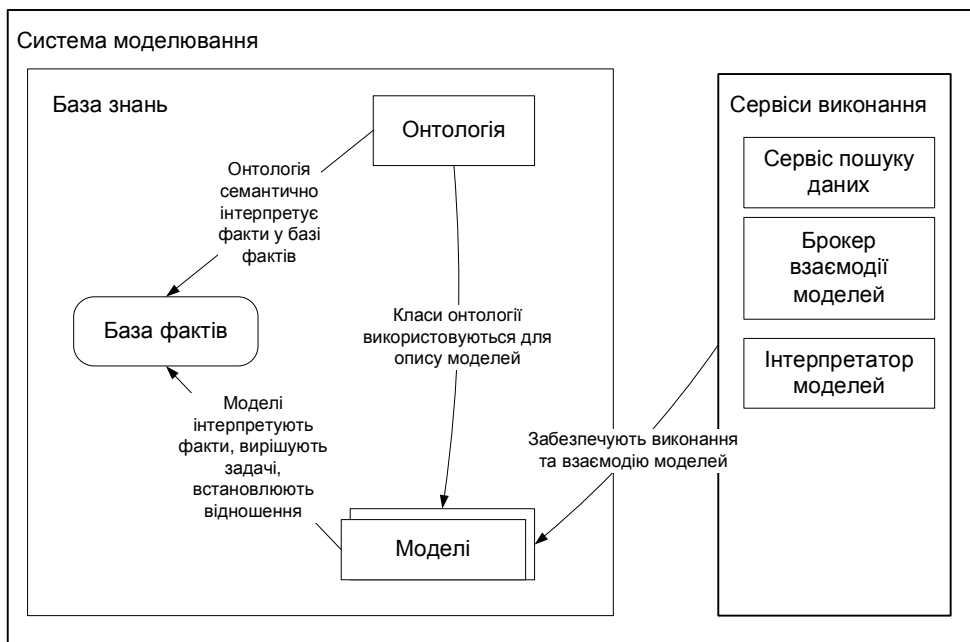


Рис. 6. Структура системи моделювання

Брокер взаємодії моделей аналізує контекст загальної моделі і визначає релевантні методи розв’язання поставленої задачі, подані іншими, детальними моделями. Для підтримки роботи брокера взаємодії моделей у системі моделювання повинна бути реалізована онтологія типів задач. З іншого боку, автори моделей формують обмеження на релевантність застосування методів, поданих моделлю у метаданих моделі. Ці обмеження брокер використовує, щоби визначити, яку конкретно модель використовувати.

### Організація підтримки та супроводу процесу верифікації моделей

Як видно зі схеми наведених моделей верифікації, процес верифікації може містити декілька етапів – завдання, які виконуються як у людино-машинному режимі за участі експертів, так і автоматично – самою системою моделювання. В інтелектуальній системі одночасно може проводитися верифікація довільної кількості моделей, при цьому для кожної задачі верифікації створюються окремі екземпляри моделі оцінювання.

Загальний нагляд над процесом верифікації моделей здійснює менеджер, відповідальний за функціонування інтелектуальної системи загалом. У випадку виявлення відхилень у роботі системи,



збоїв чи помилок, менеджер зупиняє виконання моделей. У результаті аналізу причин відхилень він може перезапустити моделі, або ініціалізувати процес їх модифікації. Власне менеджер моделей, здійснюючи нагляд над їх виконанням та маючи доступ до цілісної картини взаємодії усіх моделей інтелектуальної системи, повинен виявляти недоліки та недосконалості існуючої системи і виступати ініціатором для удосконалення моделей та процесів функціонування.

Система моделювання спрощує роботу менеджера, надаючи йому додаткову інформацію у контексті конкретного екземпляра моделі. Це стає можливим завдяки семантичній інтерпретації даних моделі, автоматичному відслідковуванню властивостей задіяних фактів та їх зв'язків з іншими фактами інформаційної бази. Так, наприклад, менеджер у процесі аналізу може переглянути схему моделі, яку рецензують, текст рецензії довільного експерта, відомості про досвід, кваліфікацію, стан здоров'я експертів та ін. Реалізація навігації за фактами інформаційної бази здійснюється на основі змістовних, семантичних залежностей між фактами, що значно спрощує роботу менеджера за рахунок відсіювання нерелевантних фактів.

### Висновок

Запропонований у роботі метод та алгоритм прийняття рішення у процесі верифікації моделі знань дасть змогу покращити якісні параметри процесу верифікації за рахунок відсіювання оцінок, що вважаються помилковими консенсусом експертів.

У роботі показано, що на відміну від простого голосування експертів запропонований метод дозволяє не тільки верифікувати модель знань, але й отримати додаткову інформацію про можливі помилки зроблені експертами у процесі верифікації моделі. Показано, що запропонований метод дає змогу виявити  $t \leq (n-1)/2$  можливих помилок експертів.

Накопичення інформації про виявлені помилки дає змогу підняти рівень аргументованості щодо складу нових груп експертів та покращити якість їх формування. Збереження історичних даних про результати верифікації моделей знань також дозволяє адекватно підбирати фахівців предметної області для розроблення нових моделей знань.

Наведена ієрархія моделей верифікації надасть системі моделювання необхідної гнучкості у виборі моделей верифікації та дасть змогу створювати інтелектуальні системи, які швидко реагують на зміни середовища.

1. *Software Bugs Cost U.S. Economy \$59.6 Billion Annually, RTI Study Finds [Electronic resource].* - Режим доступу: <http://www.nist.gov/director/prog-ofc/report02-3.pdf>.
2. Balmelli L. *Model-driven systems development/ Balmelli L, Brown D, Cantor M, Mott M.* // *IBM Systems Journal*, vol 45, Number 3, 2006. - p569 – 585.
3. *MDA Specifications. [Electronic resource].* - Режим доступу: <http://www.omg.org/mda/specs.htm>.
4. Mellor J. *Executable UML: A foundation for model-driven architecture* / Mellor S, Balcer M. - Addison Wesley, 2002. - 416p.
5. *MDA – Nice idea, shame about the... [Electronic resource].* - Режим доступу: <http://www.theserverside.com/news/1365166/MDA-Nice-idea-shame-about-the>.
6. *Mda is doa, partly thanks to soa. [Electronic resource].* - Режим доступу: [http://www.forrester.com/rb/Research/mda\\_is\\_doa%2C\\_partly\\_thanks\\_to\\_soa/q/id/39156/t/2](http://www.forrester.com/rb/Research/mda_is_doa%2C_partly_thanks_to_soa/q/id/39156/t/2).
7. Буров Є.В. *Застосування виконувальних моделей для проектування сервісно-орієнтованих інтелектуальних інформаційних систем.* / Буров Є.В. // *Вісник національного університету Львівська політехніка "Комп'ютерні науки та інформаційні технології"*, №638, 2009. – С. 200–205.
8. Буров Є.В. *Опрацювання знань у когнітивній інформаційній системі керованій моделями.* / Буров Є.В. / *Східно-Європейський журнал передових технологій.* – №6/7. – Харків, 2009. – С.40–49.
9. Barborack M. *The Consensus Problem in Fault-Tolerant Computing/ Barborack M., Malek M., Dahbura A.* // *ACM Computing Surveys.* – Vol 25, No 2, June 1993. – P. 171–220.
10. Friedman A. *System-Level Fault Diagnosis / Friedman A., Simoncini L.* // *Computer.* – 13, N 3, 1980. – P. 47–33.
11. Preparata F.P. *On the Connection Assignment Problem of diagnosable systems/ Preparata F.P., Metze G., Chien R.T.* // *IEEE Trans, on Electronic Computers.* – vol. EC-16, N 6. – 1967. – P. 848–854.
12. Pease M. *Reaching agreement in the presence of faults / Pease M., Shostak R., Lamport L.,* // *JACM*, Vol. 27, N 2. 1980. – P. 228–234.
13. Lamport L., Shostak R., Pease M. *The byzantine generals problem / L. Lamport, R. Shostak,*

*M. Pease // ACM Transactions on Programming Languages and Systems, Vol. 4, N 3, 1982. – P. 382–401.*  
14. Крамаренко М.Б. Модели диагностирования отказов параллельной вычислительной системы / М.Б. Крамаренко // *Электронное моделирование.* - № 3, 1989. – с. 60-65. 15. Гуляев В. А. Алгоритмы и методы организации процедур оперативного диагностирования в распределенных управляющих вычислительных системах / В.А. Гуляев, М.Б. Крамаренко. – К., 1988. – 55 с. (Препр. / АН УССР. Ин-т проблем моделирования в энергетике; 128).

**УДК 504.75:681.2.543**

**В. Юзевич, Н. Крап\***

Фізико-механічний інститут ім. Г. В. Карпенка НАН України, Львів

\* Львівський інститут економіки і туризму,  
кафедра природничо-математичних дисциплін

## **МОДЕЛЮВАННЯ ТУРИСТИЧНИХ ПОТОКІВ З ВИКОРИСТАННЯМ МЕТРОЛОГІЧНИХ ЧИСЕЛ**

*ã Юзевич В., Крап Н., 2011*

**Запропоновано рекомендації щодо моделювання туристичних потоків з використанням метрологічних чисел.**

**Ключові слова:** туристичні потоки, моделювання, метрологічні числа, туристичні послуги.

**The recommendation for the modelling of tourist streams with the use of metrology numbers are offered.**

**Keywords:** tourist streams, modelling, metrology numbers, tourist services.

### **Вступ**

Ринок туризму як система взаємозв'язків об'єднує велику кількість виробників турпродукції з наявними та потенційними покупцями, які мають бажання отримати якісні туристичні послуги. Від звичайних ринків товарів туристичний ринок відрізняється тим, що на ньому товар не просувається від продавців до покупців. Навпаки, покупці самі переміщуються до місця призначення, щоб отримати зарезервовані послуги.

Туристичний ринок характеризується багатоступеневістю відносин: між кінцевим споживачем і виробником туристичних послуг є посередники – туристичні агентства, туроператори, постачальники і виконавці послуг розміщення, перевезення, харчування [1].

На сучасному ринку туристичних послуг виникла необхідність у використанні метрологічних чисел для коректного забезпечення споживачів необхідною інформацією [2]. Відбір і впорядкування інформації пов'язані з аналізом непевностей (неозначеностей) [3] щодо оцінювання потоків туристів і туристичних послуг, а також з розв'язком і трактуванням результатів відповідних метрологічних задач.

**Актуальність дослідження** – метрологічні числа для дослідження стаціонарних станів і туристичних потоків у науковій літературі сфери туризму не розглядалися.

**Об'єкт дослідження** – метрологічні числа, які характеризують потоки туристів.

**Предмет дослідження** – методи удосконалення туристичних послуг з використанням метрологічних чисел, які характеризують потоки туристів.

### **Основна частина**

Метрологічні числа – це основні числа, які характеризують прикладні математичні дослідження в туристичній сфері. На відміну від чисел, в теоретичній математиці метрологічні числа можна подати як [4]: