

ШВИДКЕ РОЗРОБЛЕННЯ ЗАСТОСУВАНЬ В АДАПТИВНІЙ ТЕХНОЛОГІЇ SMARTBASE

С.Ф. Теленик, О.А. Амонс, В.С. Хмелюк, К.О. Крижова

Національний технічний університет України „КПІ”

03056, Київ, проспект Перемоги 37, корп. 18, ком. 528

e-mail: telenik@acts.kiev.ua, telenik@auts.ntu-kpi.kiev.ua, тел.: 454 9285, факс: 271 4039

У роботі розглядаються особливості реалізації застосувань на основі адаптивної технології SmartBase та метод виведення наслідків із множини нечітких правил. Запропоновано матричний метод для встановлення того, чи є певне твердження логічним наслідком системи нечітких тверджень і нечітких правил.

In given clause features of application realization on the basis of adaptive technology SmartBase and a method of a results conclusion from set of indistinct rules are considered. The matrix method for that definition is offered, whether there is some statement logical consequence of system of indistinct statements and indistinct rules.

Вступ

Автоматизація управління великими об'єктами рівня міністерств та відомств, створення програмного забезпечення яких має складну історію, вимагає нових підходів. Необхідність швидкого розроблення і впровадження системи, інтеграції існуючих застосувань, подолання недоліків алгоритмічного підходу (неприспосованість до змін і оброблення слабо формалізованих знань, неефективність у застосуваннях логічного характеру), урахування реального характеру застосувань, їх часових і надійнісних аспектів, вимагає переходу до адаптивних технологій створення інформаційно-управляючих систем [1]. Адаптивні технології, крім традиційної гнучкості, набувають надзвичайно важливих нових властивостей, таких як інтелектуалізація процесів розроблення, функціонування і розвитку інформаційно-управляючих систем (ІУС), само-організованість, широке притягнення управлінського персоналу до розвитку ІУС; зручне, без перепрограмування, настроювання системи на застосування, зміни засобів реалізації, вимог до системи тощо.

Реалізація адаптивних технологій базується на комплексному застосуванні сучасних технологій проектування, програмування, штучного інтелекту і вимагає розв'язання низки проблем. Насамперед мова йде про програмну реалізацію застосувань на основі єдиного рішення. Оскільки логічні методи найбільш застосовані для протидії динаміці прикладних областей, то необхідно створити систему логік для проектування і управління застосуваннями, розв'язання класів проблем. Особливо важливим є використання ефективних механізмів зведення в логіку за умов неповноти інформації, невизначеності, насамперед нечіткості.

Розглядаються особливості реалізації застосувань на основі адаптивної технології SmartBase та метод виведення наслідків із множини нечітких правил.

Постановка проблеми

По-перше, мова йде про комплексну проблему: необхідно розробити програмну платформу, яка забезпечує і підтримує властивості адаптивної технології. Першим кроком до її створення є побудова компонентів швидкого розроблення нових застосувань. Розробникам необхідно надати насамперед інтегроване середовище розробки нової функціональності, систему уніфікованого доступу до даних, уніфіковану систему безпеки. Бажано уже в перших версіях забезпечити сервіси контролю версій, управління розподіленою системою серверів, підтримку технології “тонкого” клієнта.

В адаптивних технологіях використовуються моделі і методи штучного інтелекту [1 – 2]. Саме вони забезпечують набуття системою здатності до адаптації. Але за умов невизначеності традиційні механізми автоматичного доведення теорем недостатньо ефективні. Водночас, резолютивний вивід довів своє право називатися базою логічного зведення.

Тому в рамках зазначеної комплексної проблеми виникає часткова проблема: необхідно розробити ефективний метод зведення наслідків у системах нечітких правил, що поєднує переваги резолютивного зведення з методами скорочення перебору варіантів, притаманними моделям роботи з невизначеністю та нечіткістю. Одним із варіантів є застосування матричного методу [3] для встановлення того, чи є певне твердження логічним наслідком системи нечітких тверджень і нечітких правил. Але матричний метод необхідно модифікувати до роботи з нечіткими оцінками.

Структура та функціонування платформи

Загальна структура системи, що функціонує на основі платформи SmartBase, показана на рис. 1. Значимо, що аббревіатури БД, СУБД, CASE використовуються для скорочення назв „база даних”, „система

управління базами даних”, „computer-aided software design”. Система в процесі функціонування і проектування базується на взаємодії серверної і декількох клієнтських частин. Після того, як ядро сервера запускається, воно обслуговує систему сервісів або перебуває у стані очікування. При запуску серверної частини автоматично завантажуються сервіси, які відповідають за створення і реєстрацію каналу зв'язку, реєстрацію класів для роботи з клієнтами. Після цього переглядається база метаданих репозитарію на наявність користувацьких застосунків або станів і ситуацій, що вимагають обслуговування.

Сервіси ядра системи – це внутрішні підсистеми, що забезпечують працездатність і функціональність системи (контроль версій, доступу до даних, безпеки і т.п.). Клієнтська частина системи забезпечує підключення до серверної частини, передбачає авторизацію користувача для конкретного застосування, запуск клієнтської частини застосування, коректне завершення сеансу.

Підсистема розроблення і розвитку побудована на ідеях концептуального моделювання, автоматизованого проектування і програмування. Для застосування ідей автоматизації програмування вводиться поняття метаданих SmartBase. Код, створений розробниками в рамках системи, тобто з використанням системи, представляється у вигляді xml-файлу, формат якого відповідає певній xslt-схемі. Метадані SmartBase в подальшому можуть бути скопійовані й одержаний код може бути виконаним. Для прикладу наведемо такі типи метаданих SmartBase: метадані – прообраз класу; метадані – прообраз збірки; метадані – прообраз застосування. База метаданих – сховище метаданих SmartBase.

Процес створення коду і його виконання в SmartBase контролюється системою. Для цього при створенні коду застосовується підсистема редагування коду, яка має графічний інтерфейс. Перед компіляцією і виконанням код контролюється підсистемою верифікації коду. Код можна зберігати не тільки у вигляді метаданих SmartBase. Виходячи із потреби в підключенні сторонніх розробок до системи у вигляді збірок, модулів тощо, код можна зберігати в базі даних і в SmartBase GAC (прообраз GAC – глобального кешу модулів). Передбачений варіант збереження збірок з машинним кодом.

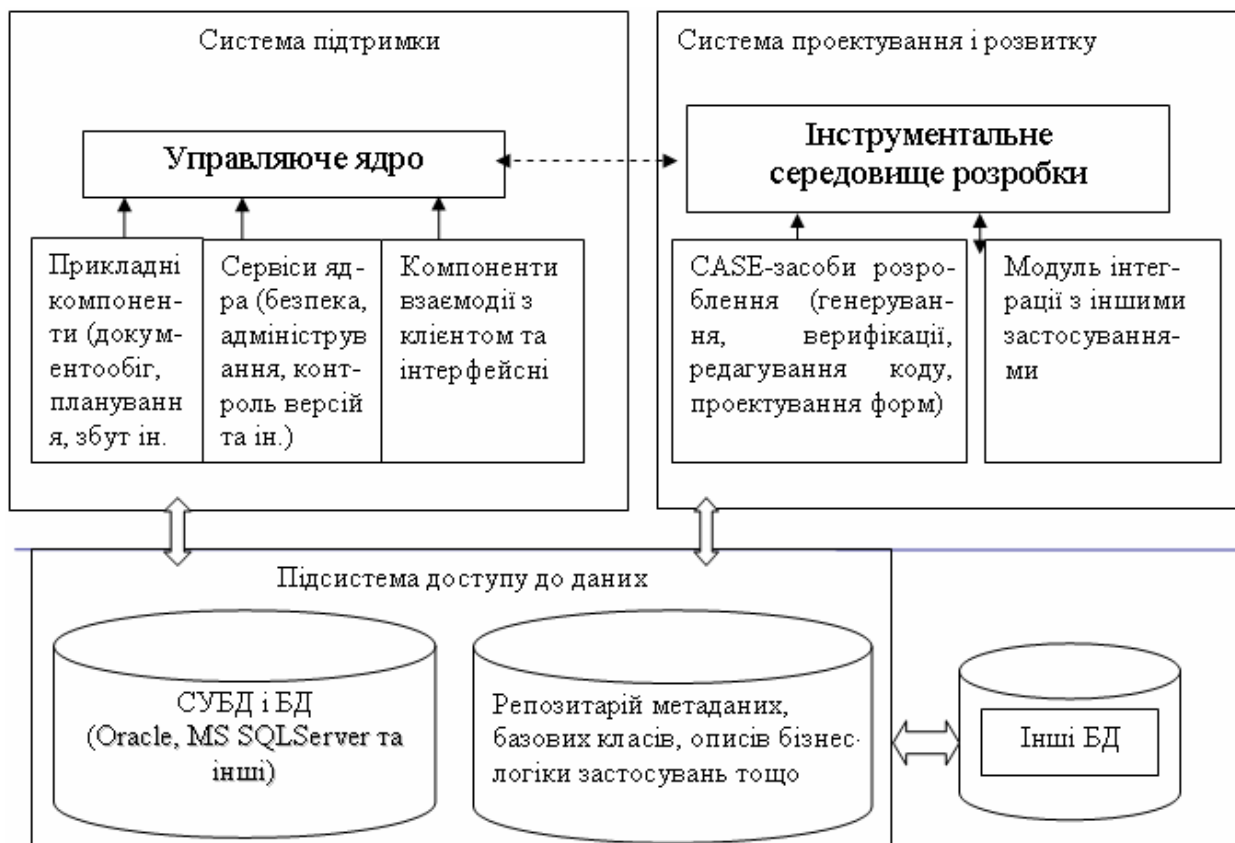


Рис. 1. Загальна структура ІУС на базі SmartBase

Для розробника в системі передбачений широкий базовий набір класів SBCL (SmartBase CLasses) – обгортка для написання стандартного коду (наприклад, створення збереженого класу зводиться до успадкування від стандартного шаблону). Застосування SmartBase – код, написаний з використанням системи (бажано) чи без, але виконуваний з використанням функцій системи. Адміністратор управляє системою, маючи ексклюзивні права на неї. Виконується авторизація користувача. Для запуску і виконання застосування SmartBase використовується система імперсонації – виконання коду від імені конкретного користувача (аж до окремих методів і доступу до атрибутів). Спеціальна утиліта перевіряє права, відповідає за коректне надання прав доступу. Передбачена база дозволів на виконання коду і доступ до даних. У системі реалізована єдина система оброблення помилок. Ефективна система контролю версій відповідає за видачу потрібної версії коду

(збірки, класу, як представлених у вигляді метаданих SmartBase, так і у вигляді файлів-збірок) і будь-яких даних.

Основними підсистемами (сервісами ядра) є підсистеми кодогенерації, верифікації та редагування коду, підсистеми безпеки, доступу до даних і контролю версій, підсистема, яка відповідає за взаємодію з клієнтами. Стисло розглянемо їх особливості.

Контроль версій. Система, яка базується на контролі версій усіх елементів застосування користувача, розроблених в системі, у які входять:

- бінарні файли (збірки, бібліотеки класів, модулі);
- ресурси користувача (документи, зображення і т.п.);
- код розширення функціональності системи.

Тонкий клієнт. Технологія, яка додатково забезпечує такі переваги застосувань, виконаних на основі платформи SmartBase:

- зникає необхідність у клієнті СУБД;
- немає потреби в бінарних файлах застосування на машині клієнта;
- відсутня проблема адміністрування клієнтської частини серверу застосувань;
- можливість доступу до серверу з будь-якого місця, де є доступ до Internet;
- можливість кешування функціональних модулів застосування на клієнтську машину.

Уніфікована система безпеки. Ця підсистема забезпечує:

- авторизацію на основі динамічних бізнес-правил;
- підтримку захищених сховищ для зберігання налаштованих параметрів, будь-яких даних, що потребують захисту від несанкціонованого перегляду та зміни;
- підтримку інфраструктури сертифікатів X509;
- шифрування клієнт-серверного трафіка.

Розподілена система серверів. Ця підсистема – основа функціонування застосувань, призначених для управління розподіленими ієрархічними об'єктами із складними вертикальними і горизонтальними зв'язками. Класичним прикладом такого об'єкта є галузь. Розподілена система серверів забезпечує:

- балансування навантаження;
- синхронізацію застосувань;
- вертикальну та горизонтальну розподіленість системи серверів.

Система уніфікованого доступу до даних. Ця підсистема уможливорює об'єктне оброблення інформації баз даних, побудованих на різних моделях. Вона забезпечує:

- незалежність розробленого в середовищі SmartBase застосування від СУБД;
- використання об'єктної моделі даних;
- генерацію структури бази даних на основі об'єктної моделі даних із SmartBase;
- генерацію об'єктної моделі даних на основі існуючої бази даних.

Нечітке логічне виведення

Однією із проблем створення застосувань є робота невизначеністю, неточністю, неповнотою інформації. Досить часто вона буває нечіткою за своєю природою. У класичних умовах застосовують логіко-математичні числення, причому механізми виведення позбавляють алгоритмічний підхід від деяких його суттєвих недоліків. Ефективним і зручним засобом для опрацювання нечітких знань є нечітка логіка. Пропонуючи апарат визначення чисельних значень нечіткості, обчислення виразів з елементами нечіткості, вона дає можливість знаходити рішення навіть у випадку недостатності даних.

Правила становлять собою вирази з імплікацією різної складності, що задаються експертами. Оскільки в цих правилах присутні нечіткі поняття, вони можуть мати більш ніж один можливий наслідок. Кожному із наслідків, в свою чергу, може бути приписана оцінка впевненості в тому, що буде мати місце саме цей наслідок. З іншого боку, маючи апарат нечіткої логіки, експерт отримує можливість формулювання правил навіть у випадку невпевненості в точному результаті. Таким чином, при розробленні адаптивних технологій маємо два аспекти, де необхідне застосування нечітких логік: оцінювання понять, які містять нечіткість; оцінювання наслідків правил.

Другий аспект є більш складним і критичним для успішного аналізу. З ним пов'язане визначення оптимальної послідовності застосування правил. Розглянемо його детальніше. Припустимо, що існує множина правил вигляду:

$$(A_1 \wedge A_2 \wedge \dots) \rightarrow (B_1 \vee B_2 \vee \dots), \quad (1)$$

де A_1, A_2, \dots – це умови правила, B_1, B_2, \dots – можливі наслідки. Для кожного наслідку експертами визначена оцінка впевненості в тому, що він має місце для виразу, який задовільняє умови правила. Наслідки правила можуть становити умови для інших правил, і таким чином ми отримуємо ланцюжки правил. Кожному наслідку визначена оцінка впевненості в інтервалі $[0,1]$. Якщо умова має один наслідок, його оцінка впевненості не обов'язково дорівнює 1.

У більшості випадків існує дві і більше можливих послідовностей застосування правил виведення. Щоб визначити, яка з них дає кінцевий результат з більшим ступенем впевненості, необхідний метод обчислення проміжних результатів на основі ступеня впевненості умови і ступеня впевненості того, що певна умова

забезпечить певний наслідок. Іншими словами, якщо відомо, що впевненість в A_1 дорівнює k , а впевненість в $A_1 \rightarrow A_2$ дорівнює l , то для подальших обчислень по ланцюжку чи (у випадку, якщо A_2 – кінцева точка) отримання кінцевої впевненості в доцільності застосування цього ланцюжка правил необхідно мати оцінку впевненості в A_2 .

Щоб отримати оцінку для A_2 , позначимо за допомогою a' “не a ”, $a \vee b$ - “або”, $a \wedge b$, “і”, $a \rightarrow b$ – імплікацію “якщо a , то b ”. В теорії нечітких множин $a' = 1 - a$. Логічному виразу “і” відповідає операція \otimes , а виразу “або” – операція \oplus . Ці операції визначаються через T-норми (позначаються знаком $*^T$) і T-конорми (позначаються знаком $+^T$) відповідно. Найпоширенішими є такі визначення T-норм і T-конорм:

$$a *^T b = ab; a +^T b = a + b - ab, \quad (2)$$

$$a *^T b = \min\{a, b\}; a +^T b = \max\{a, b\}. \quad (3)$$

Оскільки $a \rightarrow b = a' \vee b$, узагальнимо вираз імплікації на випадок нечітких множин [2]:

$$(A_1 \rightarrow A_2)(x) = A_1'(x) \vee A_2(x), \quad (4)$$

де x – конкретні умови, $A_i(x)$ – чисельні оцінки ступеня впевненості в умові/наслідку A_i .

Невідоме значення – чисельна оцінка A_2 . Позначимо:

$$A_1(x) = k.$$

$$(A_1 \rightarrow A_2)(x) = l.$$

Застосовуючи визначення операторів (2) і перетворюючи вираз, одержимо:

$$l = 1 - k + A_2(x) - (1 - k) A_2(x);$$

$$l = 1 - k + A_2(x)(1 - (1 - k));$$

$$l = 1 - k + A_2(x) k;$$

$$A_2(x) = (l - 1 + k) / k. \quad (5)$$

Оскільки формула (5) може дати від'ємні числа, якщо $(k+l) < 1$, вона не підходить для обчислення впевненості. Застосовуючи визначення операторів (3) і перетворюючи вираз, отримуємо:

$$l = \max((1 - k), A_2(x)). \quad (6)$$

Виходячи з формули (6), значення $A_2(x)$ можна обчислити підбором, який може дати більш ніж один результат. Отже, традиційним підходом задача обчислення $A_2(x)$ вирішується не однозначно і не у всіх випадках. Наведені недоліки виключаються, якщо обчислення виконувати за такою схемою:

Крок 1. Розглядати лише правила вигляду (1), в яких консеквент містить не більше одного можливого наслідку.

Крок 2. Якщо $|\rightarrow_k A$ – твердження про те, що A має місце зі ступенем впевненості k , то ступені впевненості обчислювати за допомогою формул:

$$\text{якщо } |\rightarrow_k P \text{ і } |\rightarrow_1 Q, \text{ то } |\rightarrow_{k \otimes 1} P \wedge Q;$$

$$\text{якщо } |\rightarrow_k P \text{ і } |\rightarrow_1 Q, \text{ то } |\rightarrow_{k \oplus 1} P \vee Q;$$

$$\text{якщо } |\rightarrow_k P \text{ і } |\rightarrow_1 P \rightarrow Q, \text{ то } |\rightarrow_{k \otimes 1} Q.$$

Крок 3. Перебір всіх можливих ланцюжків замінити застосуванням матричного методу виведення. На кроці 2 у випадку із двох способів визначення операцій \oplus і \otimes приймемо визначення (3). При виборі визначень (2) необхідно впевнитися, що результати не принесуть чисел, які виходять за межі діапазону $[0,1]$. Для $a *^T b = \min\{a, b\}$ найбільше можливе значення a і b дорівнює 1, і менше з них принесе 1. Це значення не виходить за межі прийнятого діапазону. Для $a +^T b = \max\{a, b\}$ найбільше можливе значення a і b дорівнює 1, і менше з них принесе 1. Це значення не виходить за межі прийнятого діапазону.

Матрична резолюція для нечіткої логіки

Невід'ємною частиною розробки алгоритму є застосування методу виведення, що дозволяє свідчити про суперечливість побудованих правил. Зокрема для клаузальної логіки найбільш простим і ефективним у даному випадку виступає матричний метод. Його основна ідея полягає у використанні матричного представлення вихідних даних і реалізації резолютивного методу у вигляді операції множення матриць. Такі матриці, очевидно, повинні мати стовпець і рядок для кожного літерала, а елемент на перетині рядка i та стовпця j буде мати ненульове значення лише тоді, коли знайдеться клауза, у безумовній частині якої знаходиться літерал рядка i , а умовна частина містить літерал стовпця j . Доведено, що множина клауз S суперечлива тоді і тільки тоді, коли застосування процедури множення таких матриць приводить до побудови матриці M^p з ненульовою головною діагоналлю. Наприклад, нехай маємо множину клауз: $A \leftarrow B$; $B \leftarrow C$; $C \leftarrow$ і необхідно обґрунтувати, що A є їх логічним наслідком. Додавляємо клаузу $\leftarrow A$. Нагадаємо, що запис клауз традиційно міняє місцями умовну і безумовну частини. Зрозуміло, що порожню клаузу можна отримати за допомогою трьох застосувань правила резолюції. Множення матриць підтверджує цей висновок. Отже, маємо:

$$\begin{array}{cccc}
 \diamond & A & B & C \\
 \diamond & 0 & 0 & 0 & 1 \\
 M = A & 1 & 0 & 0 & 0, M^2 = A & 0 & 0 & 0 & 1, M^3 = A & 0 & 0 & 1 & 0, M^4 = A & 0 & 1 & 0 & 0. \\
 B & 0 & 1 & 0 & 0 & B & 1 & 0 & 0 & 0 & B & 0 & 0 & 0 & 1 & B & 0 & 0 & 1 & 0 \\
 C & 0 & 0 & 1 & 0 & C & 0 & 1 & 0 & 0 & C & 1 & 0 & 0 & 0 & C & 0 & 0 & 0 & 1
 \end{array}$$

З огляду на суперечливість множини клауз можлива ознака суперечливості у матричному варіанті – наявність ненульової головної діагоналі у матриці M^p , де $p = 4$, – підтверджується. При побудові матриці застосовується два правила:

Правило А. Якщо в матриці є нульовий рядок, то вилучаємо його з матриці разом з відповідним стовпцем.

Правило В. Якщо в матриці є порожні стовпці, то вилучаємо відповідні клаузи і після цього застосовуємо правило А.

Застосуємо вищезазначений метод виведення для встановлення того, що ступінь впевненості потрібного висновку є ненульовим. Розглянемо приклад оцінок експертів, повертаючись до традиційного запису:

- $\mapsto_{0.7} A_1$
- $\mapsto_{0.4} (A_1 \rightarrow A_3)$
- $\mapsto_{0.5} ((A_3 \wedge A_4) \rightarrow A_5)$
- $\mapsto_{0.7} A_2$
- $\mapsto_{0.5} (A_2 \rightarrow A_4)$
- $\mapsto_{0.3} ((A_3 \wedge A_4) \rightarrow A_6)$

Спочатку застосуванням визначення операцій (3) і правил виведення кроку 2 вище наведеної схеми:

- 1) обчислимо $\mapsto_{0.7 \otimes 0.4} A_3$:
 $0.7 \otimes 0.4 = \min\{0.7, 0.4\} = 0.4$,
 тобто $\mapsto_{0.4} A_3$;
 - 2) обчислимо $\mapsto_{0.7 \otimes 0.5} A_4$:
 $0.7 \otimes 0.5 = \min\{0.7, 0.5\} = 0.5$,
 тобто $\mapsto_{0.5} A_4$;
 - 3) обчислимо $\mapsto_{0.4 \otimes 0.5} (A_3 \wedge A_4)$:
 $0.4 \otimes 0.5 = \min\{0.4, 0.5\} = 0.4$,
 тобто $\mapsto_{0.4} A_3 \wedge A_4$;
 - 4) обчислимо $\mapsto_{0.4 \otimes 0.5} A_5$:
 $0.4 \otimes 0.5 = \min\{0.4, 0.5\} = 0.4$,
 тобто $\mapsto_{0.4} A_5$;
 - 5) обчислимо $\mapsto_{0.4 \otimes 0.3} A_6$:
 $0.4 \otimes 0.3 = \min\{0.4, 0.3\} = 0.3$,
 тобто $\mapsto_{0.3} A_6$;
- Отже,
 $\mapsto_{0.4} A_5, \mapsto_{0.3} A_6$.

Тепер застосуємо матричний метод. Спочатку побудуємо матрицю для перевірки висновку A_5 . Наявність оцінок впевненості правил і тверджень вимагає вдосконалити процедуру побудови матриць. Вони, як і для двозначної логіки, повинні мати стовпець і рядок для кожного літерала, а елемент на перетині рядка i та стовпця j буде мати ненульове значення лише тоді, коли знайдеться клауза, у безумовній частині якої знаходиться літерал рядка i , а умовна частина містить літерал стовпця j . Але це значення буде нечіткою оцінкою. Додаємо клаузу $\leftarrow A_5$. З огляду на можливість існування декількох клауз із однаковою безумовною частиною і клауз із умовною частиною з декількома літералами (наприклад, порівняйте $\mapsto_{0.4} A_1, \mapsto_{0.6} (A_2 \rightarrow A_3), \mapsto_{0.2} (A_1 \rightarrow A_3), \mapsto_{0.7} A_2 \wedge \mapsto_{0.4} A_1, \mapsto_{0.7} A_2, \mapsto_{0.8} (A_1, A_2 \rightarrow A_3)$), наявність першого випадку будемо позначати індексом „ \vee ” біля кожного літерала умовної частини однієї клаузи, а наявність останнього випадку – індексом „ \wedge ” біля кожного літерала умовної частини однієї клаузи. Матриця має вигляд

\diamond	A1	A2	A3	A4	A5	A6
\diamond	0	0	0	0	1	0
A1	0,7	0	0	0	0	0
A2	0,7	0	0	0	0	0
A3	0	0,4	0	0	0	0
A4	0	0	0,5	0	0	0
A5	0	0	0	0,5 \wedge	0,5 \wedge	0
A6	0	0	0	0	0	0

Застосовуючи правила А та В приведемо матрицю до вигляду матриці M1. Модифікуємо також алгоритм множення матриць. У випадку наявності правил $\mapsto_k A_1, \mapsto_l (A_1 \rightarrow A_2)$, множення $k \otimes l$ необхідно виконувати за формулою $\min\{k, l\}$. У випадку наявності правил $\mapsto_k A_1, \mapsto_l A_2, \mapsto_q (A_1, A_2 \rightarrow A_3)$, у матриці у рядку з’являються індекси „ \wedge ” у стовпцях для літералів A_1, A_2 і множення $k \otimes l$ необхідно виконувати за формулою $\min\{\min\{k, q\}, \min\{l, q\}\}$. У випадку наявності правил $\mapsto_k A_1, \mapsto_l A_2, \mapsto_q (A_1 \rightarrow A_3), \mapsto_r (A_2 \rightarrow A_3)$, у матриці у рядку з’являються два ненульових елементи з індексами „ \vee ”, у стовпцях для літералів A_1, A_2 і при множенні необхідно вибрати максимальний із двох добутків у рядку $\max\{\min\{k, q\}, \min\{l, r\}\}$.

	\diamond	A_1	A_2	A_3	A_4	A_5
\diamond	0	0	0	0	0	1
A_1	0,7	0	0	0	0	0
A_2	0,7	0	0	0	0	0
A_3	0	0,4	0	0	0	0
A_4	0	0	0,5	0	0	0
A_5	0	0	0	0,5 \wedge	0,5 \wedge	0

Матриця M_1

	\diamond	A_1	A_2	A_3	A_4	A_5
\diamond	0	0,4	0,5	0	0	0
A_1	0	0	0	0,5 \wedge	0,5 \wedge	0
A_2	0	0	0	0,4 \wedge	0,5 \wedge	0
A_3	0	0	0	0	0	0,4
A_4	0	0	0	0	0	0,4
A_5	0,4	0	0	0	0	0

Матриця M_1^3

	\diamond	A_1	A_2	A_3	A_4	A_5
\diamond	0	0	0	0,5 \wedge	0,5 \wedge	0
A_1	0	0	0	0	0	0,7
A_2	0	0	0	0	0	0,7
A_3	0,4	0	0	0	0	0
A_4	0,4	0	0	0	0	0
A_5	0	0,4 \wedge	0,5 \wedge	0	0	0

Матриця M_1^2

	\diamond	A_1	A_2	A_3	A_4	A_5
\diamond	0,4	0	0	0	0	0
A_1	0	0,5 \wedge	0,5 \wedge	0	0	0
A_2	0	0,4	0,5	0	0	0
A_3	0	0	0	0,4 \wedge	0,4 \wedge	0
A_4	0	0	0	0,4 \wedge	0,4 \wedge	0
A_5	0	0	0	0	0	0,4

Матриця M_1^4

Перемноження матриці M_1 призводить до підтвердження суперечливості вихідної множини клауз при $p = 4$ за ознакою суперечливості для традиційної клаузальної логіки [3], що доводить вищенаведене припущення про логічний наслідок A_5 . Залишається визначитися з нечіткою оцінкою результату. Оскільки застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми ми одержали $\mapsto_{0.4} A_5$, то нечітка оцінка результату належить діагональному елементу останньої матриці на перетині рядка і стовпця A_5 . Перейдемо до узагальнення. Спочатку опишемо власне процедуру *Matrix-fuzzy-deduction* нечіткого виведення на основі матричного методу:

Крок 1. Будуємо матрицю M для вихідної множини нечітких правил.

Крок 2. Застосовуємо правило A до матриці M .

Крок 3. Застосовуємо правило B до матриці M .

Крок 4. За необхідності повертаємося на крок 2, інакше $p = 1$ і переходимо на крок 5.

Крок 5. Отримуємо матрицю M^p , модифікуючи операцію множення матриць: у випадку наявності індексів „ \wedge ” у стовпцях для літералів A_1, A_i множення ($k \otimes$) $\otimes q$ необхідно виконувати за формулою $\min\{\min\{k, q\}, \min\{l, q\}\}$; у випадку наявності індексів „ \vee ” у стовпцях для літералів A_1, A_2 для множення необхідно вибрати максимальний із двох добутків у рядку $\max\{\min\{k, q\}, \min\{l, r\}\}$. Якщо виконується ознака суперечливості або супротивного випадку, то кінець, інакше $p = p + 1$ і переходимо на крок 5.

Оскільки ознака виводимості – наявність певної структури C на головній діагоналі – сформульована, залишається довести правомірність її застосування.

Теорема 1. Множина нечітких правил S суперечлива тоді і тільки тоді, коли застосування процедури *Matrix-fuzzy-deduction* приводить до побудови матриці M^p з ненульовою головною діагоналлю. При цьому нечітка оцінка належить діагональному елементу останньої матриці на перетині рядка і стовпця, які відповідають нечіткому твердженню, що перевіряється, і відповідає оцінці, отриманій застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми.

Доведення. Пряма теорема. Нехай застосуванням процедури *Matrix-deduction* встановлено існування такого натурального p , що матриця M^p має ненульову головну діагональ. Доведемо, що множина клауз суперечлива.

По-перше, доведемо, що правило A залишає суперечливу множину клауз суперечливою. Дійсно, якщо множина клауз суперечлива, то існує послідовність резолюцій, останньою резолювентою якої є порожня клауза. З визначення правила резолюції негайним наслідком є те, що в зазначеній послідовності не може приймати участь клауза, в яку входить вилучений за допомогою правила A літерал, адже відповідний літерал не зустрічається в прaviх частинах клауз. Тоді застосування правила A не може привести до зникнення зазначеної послідовності, тобто множина клауз залишається суперечливою.

По-друге, доведемо, що правило B залишає суперечливу множину клауз суперечливою. Дійсно, якщо множина клауз суперечлива, то існує послідовність резолюцій, останньою резолювентою якої є порожня клауза. З визначення правила резолюції негайним наслідком є те, що в зазначеній послідовності не може приймати участь вилучена за допомогою правила B клауза, адже відповідний літеральне не зустрічається в лівих частинах клауз. Тоді застосування правила B не може привести до зникнення зазначеної послідовності, тобто множина клауз залишається суперечливою.

Нехай застосування правил A, B не привело до встановлення несперечливості множини клауз і побудовою матриць M^p встановлено існування такого натурального p , що матриця M^p має ненульову головну діагональ. Доведемо, що множина клауз суперечлива.

Будемо використовувати індукцію за максимальною резолютивною відстанню (максимальною довжиною від літерала у правій частині кляузи до порожнього літерала у лівій частині кляузи через резолютивні переходи). Тоді базис індукції пов'язаний з правилом $\vdash_k A_1$ і встановленням того, що A_1 є логічним наслідком цієї кляузи. Очевидно, що в цьому випадку множина кляуз суперечлива, а нечітка оцінка належить діагональному елементу для A_1 і відповідає оцінці, отриманій застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми. Індуктивне припущення: теорема виконується для довільної системи кляуз з максимальною резолютивною відстанню n . Залишається довести, що теорема виконується для довільної системи кляуз з максимальною резолютивною відстанню $n+1$. Дійсно, відома теорема про відповідність циклів і ненульових елементів головної діагоналі матриці суміжності графа на основі індуктивного припущення свідчить про заповнення всіх діагональних елементів для всіх шляхів довжини n . Тоді ще одне припущення матриць приводить до заповнення всіх інших діагональних елементів. При цьому визначення операцій через мінімальні і максимальні елементи належить діагональному елементу останньої матриці на перетині рядка і стовпця, які відповідають нечіткому твердженню, що перевіряється, і відповідає оцінці, отриманій застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми.

Обернена теорема. Нехай множина нечітких правил суперечлива і застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми отримана нечітка оцінка. Доведемо, що застосування процедури *Matrix-fuzzy-deduction* приводить до побудови такої матриці M^p , що її головна діагональ ненульова, а діагональний елемент для твердження, що перевіряється, містить нечітку оцінку, яка відповідає оцінці, отриманій застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми. Застосування правил A, B не змінить суперечливість множини правил. За умови суперечливості множини правил існує така послідовність застосувань правила резолюції, що остання резолювента є порожньою. Тоді за побудовою матриця M^1 містить шляхи довжини 1 від літерала в рядку до літерала в стовпці. Тоді із визначення множення матриць M^2 містить шляхи довжини 2 від літерала в рядку до літерала в стовпці. В загальному випадку M^p містить шляхи довжини p від літерала в рядку до літерала в стовпці.

Якщо всі застосування правил A, B відсікають літерали, які не приймають участь у резолюціях, то тоді залишаються лише літерали, які безпосередньо приймають участь у резолюціях. Отже, всі діагональні елементи повинні врешті решт стати ненульовими, а ті які вже стали ненульовими, не перетворяться у нульові. При цьому визначення операцій через мінімальні і максимальні елементи повторює резолюції, а діагональний елемент останньої матриці на перетині рядка і стовпця, які відповідають нечіткому твердженню, що перевіряється, буде відповідати оцінці, отриманій застосуванням визначення операцій (3) і правил виведення кроку 2 вищенаведеної схеми. *Теорема доведена.*

Висновки

Застосування першої версії платформи SmartBase підтвердило працездатність закладених в її основу ідей. З її використанням швидко створено інформаційно-керуюче ядро та декілька застосувань системи управління одного з міністерств. На базі платформи, наприклад, ефективно функціонує комплекс електронного діловодства, документообігу і контролю рішень. Розвиток платформи пов'язаний із залученням алгоритмів інтелектуального аналізу даних і адаптації, схем розрахунків. Запропонована нова стратегія резолютивного виведення для нечітких правил на основі застосування матричних форм дозволяє застосовувати механізми виведення за умов нечіткої інформації. Подальший розвиток досліджень пов'язаний з розробкою методик відтворення виводів за послідовністю результатів множення матриць а також застосуванням методу для нечітких нейромереж.

1. Павлов А.А., Теленик С.Ф. Адаптивные технологии и алгоритмизация в системах управления. – Киев: Техника, 2003. – 340 с.
2. Ковальски Р. Логика в решении проблем. – М.: Наука, 1990. – 280 с.
3. Теленик С.Ф., Амонс О.А., Смічик Р.В., Хмелюк В.С. Матрична резолюція для кляузальної логіки // Вісник ХНАДУ, 2005. – вип. 22. – С. 94 – 97.
4. Nguyen H.T, Walker E.A. A First Course in Fuzzy Logic. – CHAPMAN & HALL/CRC. – 2000.