

УДК 004.45:007.51

ОНТОЛОГІЧНИЙ АНАЛІЗ СППР

Ю.В. Бондарчук

Київський національний університет імені Тараса Шевченка

e-mail: byv.kyiv@gmail.com

СППР є таким програмним/апаратним рішенням, яке з'єднує методи прийняття рішень з рішенням(и) в конкретній прикладній області [1]. І хоча реалізованої СППР *взагалі* не може бути, проте є уявлення та відношення до такого явища *взагалі*, яке вирізняє його серед інших програмних систем. Представлення такого (загального) розуміння особливостей на високому рівні абстракції і є онтологією. Від вибору онтології, як мінімум, отримуємо вплив на таксономію та прагматичні аспекти СППР. Щодо СППР, на відміну від інших категорій програмних систем, реалізація більше розвинута, а щодо абстракції часто користуються чисто інтуїтивним розумінням – головне, щоб там фігурували методи прийняття рішень.

Пропонується онтологію СППР представляти із таких об'єктів та їх підпорядкування як на рис. 1 (стрілки без адресата – можливість інших елементів). До онтології

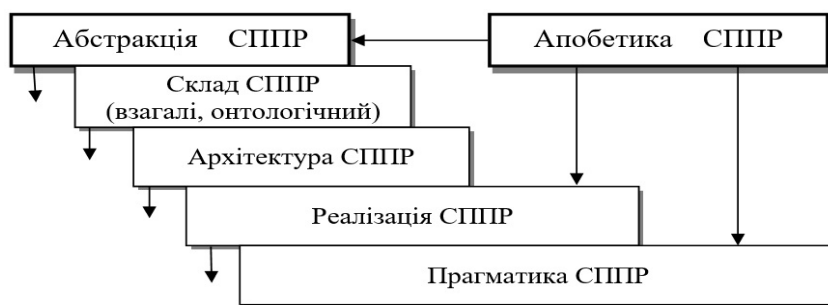


Рис. 1

введена апобетика (тобто *задум, мета* розробника, також запит цілої галузі чи суспільства), яка в ІТ-проектах *взагалі* фігурує одним абзацом, проте «розгортка» апобетики є основою абстракції, реалізації та прагматики СППР. На практиці апобетику підміняють прагматичними аспектами програмних рішень. (Наприклад, аналітична система може, крім своєї заявленої функції, просто слідкувати за користувачем, його інтересами та, в кращому разі, таргетувати під нього рекламу. Тобто очікувана дія користувача, прагматика, та бажаний результат, апобетика, зовсім різні – і система не є аналітичною системою. Аналогічно багато СППР за назвою не є СППР, а процедурами імплементації певних методів прийняття рішень.) Якщо в апобетику включаємо вимогу до СППР оперування *рішенням(и)* як *самостійним об'єктом* [1, 2], то і абстракція, склад, архітектура та реалізація отримають відповідне представлення, що вплине на прагматику (як очікувану дію).

Абстракцію СППР пропонується розглядати на рівні наступних компонентів (рис. 2) [2]. Пропонується оперування рішеннями як об'єктами, які *можуть зберігатися*,

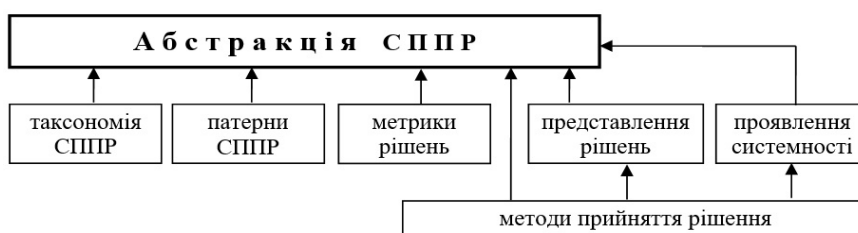


Рис. 2

повторно використовуватися та передаватися іншим СППР, вважати основною властивістю СППР та основним проявленням системності. Це дає можливість вводити адекватну таксономію (розкриваючи властиву тільки цьому класу систем особливості) [3]. Відповідно, з'являється аналіз типових операцій з рішеннями у вигляді *патернів* СППР (для маніпуляції з рішеннями як з окремими об'єктами) [2] та метрик рішень [4] (представлення складності та корисності СППР).

Із запропонованого проявлення системності СППР можна сформулювати критерій технологічної досконалості (або зрілості) СППР: СППР має бути лише оболонкою (середовищем) оперування з рішенням ОПР та експорту/імпорту з інших СППР. Іншими словами – це степінь відокремлення рішення від СППР, яка є лише конструктором рішень. І максимальне відокремлення і є максимальна досконалість. До технічної досконалості пропонується [5] віднести наявність: 1) обміну рішеннями; 2) реакції СППР на зовнішні запити; 3) розпізнавання подібних систем для координації дій з підтримки прийняття рішень; 4) імпорту сценаріїв розподілених дій; 5) повний життєвий цикл рішення (як окремого об'єкту).

Пропонується для онтологічного розгляду склад СППР (мінімальний, базовий, лише обов'язкові складові) як на рис. 3. Це 4 елементи: ОПР, СППР, рішення та

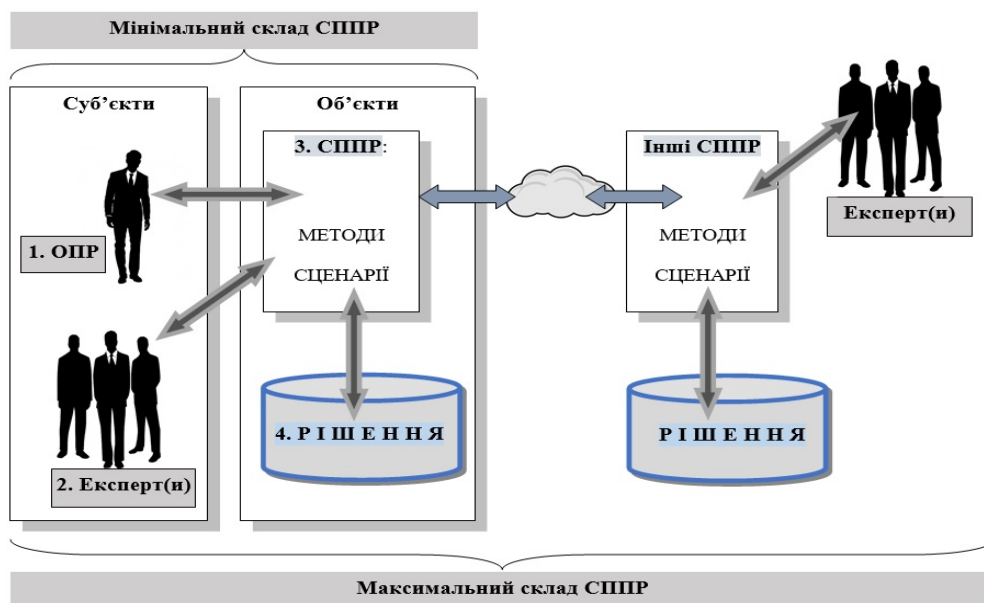


Рис. 3

експерт(и). Рішення, як бачимо, винесені із СППР, хоча в реалізації вони в СППР, яка і відтворює сценарій роботи у формуванні рішення. Остання є оболонкою (сценарієм) об'єднання методів з рішенням(и) чи отримання рішення. Програмне рішення із цих 4-х елементів і є правильна СППР. Як бачимо, тут не вдається уникнути подвійного використання терміну СППР, в якому є окремо елементом СППР, але так вже склалося: ОПР (обов'язково) та експерт(и) (як правило) є в «контурі» СППР і конкретна програмна система також є СППР. Відокремлення рішень (для окремих дій з ними) робить СППР (як явище в ІТ) реально унікальним. Бо аналог ОПР є в будь-якій інтерактивній системі, також експерти часто задіяні в аналітичних системах, - але тепер ми їх не плутаємо, вибравши адекватну онтологію. На думку автора, така онтологія все спрощує серед маси схожих проявлень в сфері ІТ. Більше того, певні нові реалізації в сфері «штучного» інтелекту впевнено можна віднести до СППР (там, де формується рішення чи модифікується із бази рішень, враховує експертну думку та адаптується до ОПР, користувача; та замість «традиційних» класичних та важких для онлайн реалізацій малокорисних методів

прийняття рішень застосовується інтелектуальний аналіз даних, машинне навчання тощо). Схоже, що якраз в сфері «штучного» інтелекту і з'явилися дійсно правильні (на думку автора) СППР.

Отже, мінімальний («онтологічний») склад СППР є із 4-х елементів, які можуть розширюватися до максимального складу (рис. 3) через взаємодію СППР з іншими СППР та їх експертами. Але нарощування складності відбувається лише базовими 4-ма елементами. Звернемо увагу, що тут немає даних (хоча СППР є програмами), бо даними для СППР буквально є *рішення*, а вже методи прийняття рішень потребують даних. Тут є певна аналогія з функціональним програмуванням – нас цікавить *що має робитися*, а не скільки *як*. Також аналогією може бути шаблон веб-документу, який при відкритті в браузері стане реальною сторінкою (із свіжими новинами, пропозиціями тощо). Цей веб-шаблон є універсальним представленням (вікном у дані) для користувачів. І цей шаблон, яким би він не був складним, можна вмістити у фрейм в іншому документі. Отака однорідність має бути ідеалом для рішень як самостійних об'єктів.

Як бачимо, онтологічний склад СППР вже вносить ясність (у порівнянні з іншими програмами) та допомагає виявляти унікальну системність СППР. Але щодо базових елементів, якщо не вказати їх атрибути в онтології, то доведеться оперувати лише інтуїтивним уявленням про роль кожного із елементів. Крім того, саме наявність атрибутів (ознак) дозволяє предметно класифікувати, порівнювати СППР. Для СППР вибраний підхід, який використовується в представленні референсних (типових) корпоративних архітектур.

Пропонується базові елементи наділити атрибутами як на рис. 4. Для них спеціально вибрані назви, які точніше передають специфіку атрибуту (наприклад, замість



Рис. 4

прагматика, яка різними спеціалістами (вченими, розробниками, продавцями) може розумітися дуже вузько, використовується *очікувана дія*). Такі атрибути мають бути не «технологічними», а сутнісними. Так *носій* (певна форма) має бути в реалізації, але будучи єдиним фізичним представленням, може мати багато спільного з багатьма іншими (не СППР) системами. Тому аналізувати системи у їх фізичному представленні найменш конструктивна справа (вище наводився приклад, коли СППР схожа на системи, які точно не СППР). *Кодування* (мова) в СППР це не мови/технології програмування систем, це є мова(и) опису рішення та маніпуляції з ними (конструювання рішень). *Вкладена ідея* (аналог семантики) знову ж стосується формування суті рішення та ролі в цьому процесі ОПР, СППР як оболонки сценарію, бази рішень та експертів. На рівні вкладеної ідеї з'єднується внутрішня реалізація з предметною областю: наскільки ідея дій буде розкрита для ОПР та експертів, настільки буде оцінена низько/високо зручність роботи із СППР. Тут формуються та розкриваються функціональні можливості СППР, зокрема, характерна саме для СППР функція пояснення дій системи при формуванні рішення. *Очікувана дія* (аналог прагматики) кожного із базових елементів стосовно формування рішення (всі інші моменти прагматики є загально програмними). *Бажаний результат* (реалізація

апогетики) – зокрема, поведінка кожного із базових елементів в шкалі від мінімальної до максимальної технологічної досконалості СППР.

Для розробників СППР заповнити таблицю на рис. 4 не буде простою задачею, бо те, що звикли залишати на рівні інтуїтивного розуміння потрібно представити в конкретних поняттях. Зате зразу «продивляються» варіанти таксономії та родинна спільність СППР чи аномалії реалізації. Крім того, в СППР з'являється не лише досить розмитий «фасад» - ми легко виявляємо масу особливостей, притаманних лише СППР.

Отже, базові елементи отримали вище наведені атрибути. Але ж ці 4-и елементи мають також *локалізацію*. Останню не варто відносити до сутнісних атрибутів, а розглядати окремо. Внесення локалізації в онтологічний аналіз також конкретизує апарат таксономії та порівняння СППР. На рис. 5 пропонується варіант локалізації базових елементів.



Рис. 5

Перехід від локальних СППР до хмарних (типу SaaS, наприклад) завершить формування технологічної досконалості СППР через потребу в уніфікації представлення рішень (типу референсних шаблонів рішень), протоколів їх конвертації та обміну. В ідеалі СППР мають стати «браузерами» роботи ОПР з даними, методами прийняття рішень, базами (хмарами) рішень та експертами.

Література

1. Бондарчук Ю.В. Щодо модифікації парадигми СППР //Системи підтримки прийняття рішень. Теорія і практика (СППР-2013): Збірник доповідей наук.-практ. конф. – Київ: ІПММС НАНУ, 2013. - С. 9-11.
2. Бондарчук Ю.В. Патерни взаємодії СППР //Системи підтримки прийняття рішень. Теорія і практика (СППР-2015): Збірник доповідей наук.-практ. конф. –Київ: ІПММС НАНУ, 2015. - С. 9-12.
3. Бондарчук Ю.В. Варіант таксономії СППР //International conference "Problems of decision making under uncertainties" (PDMU-2014).). 12-16.5.2014, Мукачево. - С. 68-69.
4. Бондарчук Ю.В. Метричні характеристики розподілених СППР //International Workshop "Problems of decision making under uncertainties" (PDMU-2013). 13-17.5.2013, Східниця. - С. 93-94.
5. Bondarchuk Y.V. About technological perfection of DSS // XXVI International conference "Problems of decision making under uncertainties" (PDMU-2016). 23-27.5.2016, Tbilisi-Batumi, Georgia. - p. 26.