

УДК 004.45:007.51

ЩОДО МОДИФІКАЦІЇ ПАРАДИГМИ СППР

Ю.В. Бондарчук

Київський національний університет імені Тараса Шевченка

e-mail: byv@univ.kiev.ua

На думку автора, виродженню великих сподівань щодо СППР (як спеціального окремого класу програмних систем) у спеціалізовані інформаційно-пошукові системи пов'язано з:

- вкрай вузькою спеціалізацією, що заважає СППР проявитися як окремому класу систем, - СППР, як явище, скоріше деградує;
- закладеною розробниками вродженою неможливістю СППР взаємодіяти між собою (навіть однотипних, але різних власників);
- відсутністю прототипів настільки розвинутих, щоб їх спосіб представлення сценаріїв та формальний опис рішень став би одним із стандартів профілю протоколів СППР;
- відсутністю мереж взаємодіючих СППР – розподілених СППР (як наслідок відсутності профілю протоколів), спочатку однотипних, але далі буде показано, що будь-які СППР можуть взаємодіяти;
- розвитком пошукових порталів в Інтернеті, які створили ілюзію, що в Інтернеті є все для прийняття рішень. Проте неструктурований опис рішень (взагалі експертних думок, оцінок) не дає можливості автоматизувати їх обробку;
- сподіванням на прорив у задачах теорії ігор, але, враховуючи їх складність, коли для практичних задач потрібний неприпустимо великий час обчислень (і тоді чи це є потрібна теорія?), хто буде користуватися СППР з надскладним підбором варіантів, коли необхідно приймати рішення зараз?
- врешті, таким станом помилкового уявлення, що все програмно-інформаційне, яке використовувалося під час формування рішень, "отримало" право називатися якщо не системою, то процедурою прийняття рішень. Тобто будь-який довідничок також може претендувати на підсистему прийняття рішень.

Враховуючи безпосереднє призначення СППР – бути методологічною та інформаційною підтримкою ОПР в прийнятті рішень, пропонується, зокрема, СППР обов'язково наділити функцією *бути також системою-сховищем вже прийнятих рішень*. СППР для своєї предметної області *має містити набір рішень* (попередньо узагальнений розробниками, експертами тощо). А експлуатацією СППР еволюціонує надалі *розвитком вже існуючих та створенням нових рішень*.

Чи не логічно в СППР, змінивши дещо сучасну парадигму (застою), поступити за аналогією типового життєвого методу роботи з рішеннями – опертися на вже існуючі рішення? Наприклад, як із кулінарією, більшість блюд (=приміненнь) у харчуванні (=задача) не потребують винайдення рецептів (=прийняття рішень), а потрібно лише розібратися в існуючих (=почати з будь-якого допустимого рішення). Типовий кулінарний рецепт є лише шаблоном, який можна самостійно "масштабувати".

Об'єктне програмування/архітектура взагалі базується на шаблонах (класах) поведінки даних та методів. В СППР потрібно наважитись на щось подібне! Ну не СППР же приймає рішення за ОПР? Але ОПР не вистачає інструментів:

- опису рішень (взагалі будь-яких в "термінах рішень"), в ідеалі якби це були класи із атрибутів, методів та даних з підтримкою версій;

- порівняння рішень (тут, крім формальних процедур, нарешті станемо потрібними ми - експерти);
- опису сценаріїв прийняття рішень, коли СППР могла б бути, як мінімум, інтерпретатором сценарію.

Вище перераховані три інструменти дозволяють *накопичувати* бази рішень (моделей, шаблонів, сценаріїв, оболонок, обгортки тощо), *обмінюватися* ними, *використовувати* практично в будь-якій СППР (готовій до цього). І в подальшій перспективі – створення інтернет-сервісу прийняття рішень.

Типова кібернетична модель СППР (рис. 1) не пропонує варіант рішення і не веде базу рішень.

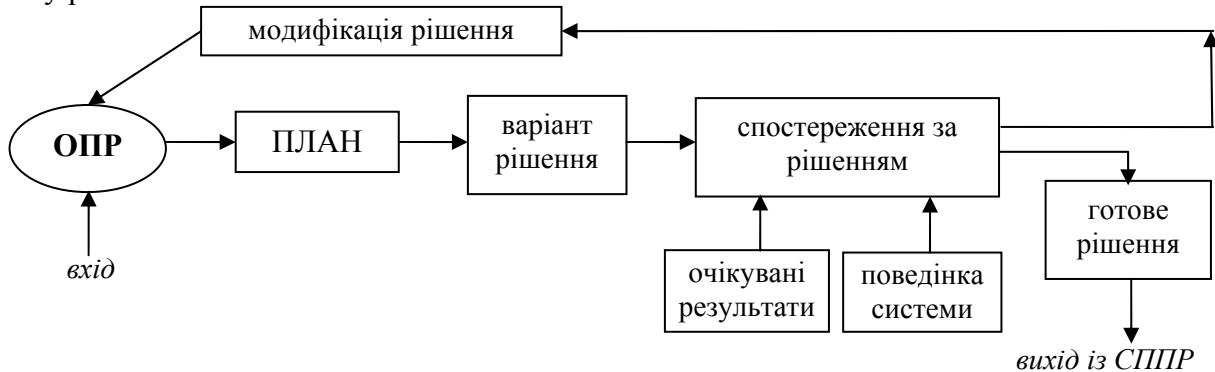


Рис. 1

Якщо дивитися на СППР як на вже організовану множину рішень, то ще більше "підсилюємо" кібернетичну модель СППР (рис. 2).

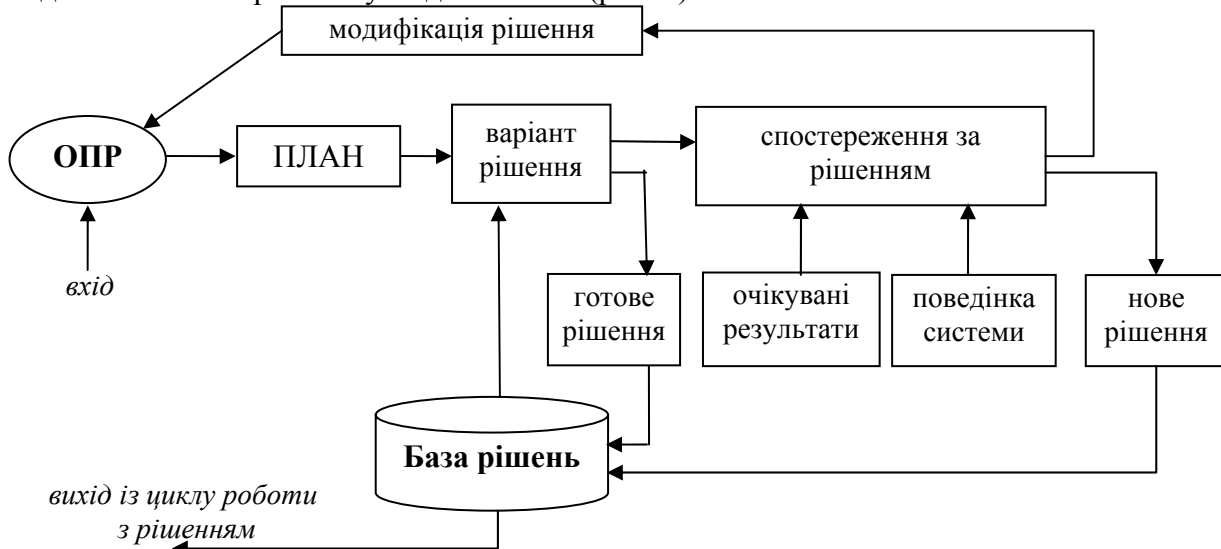


Рис. 2

В новій моделі: 1) на початку циклу прийняття рішення СППР *вже пропонує* варіант рішення, можливо навіть простішої задачі; 2) кожне нове рішення *поповнює базу рішень*, а повторне використання наявного там варіанту також певним чином відмічається. Отже, ОПР має розуміти СППР так: *можу взяти готове рішення* (можливо, спочатку найгірше), а *можу створити нове рішення*. Зрозуміло, що складність починається з боку архітектури (реалізації). Тут головне виробити спосіб опису рішень, що дозволить їх накопичувати, порівнювати (хоча б через експертів). А реалізація розподілених СППР стає наступним природним кроком.

Для фахівців із СППР потрібно бачити, що розвиток пошукових систем має вступити в фазу набуття рис СППР. Зараз це вже частково є присутнім, якщо за ключовою фразою можна перейти на спеціалізований ресурс, який "в темі". Така тимчасова форма позбавлена оберненого зв'язку роботи з рішенням. А без циклу:

план рішення → впровадження рішення → модифікація рішення
засобами самої СППР – це ще не СППР.

Без створення розподілених СППР і згодом переміщення СППР в хмарні сервіси годі сподіватися зупинки процесу сповзання СППР із систем у процедури прийняття рішень. В множині розподілених СППР (тобто взаємодіючих через мережу) окрема СППР може приймати участь в якості:

- джерела рішення (є варіант чи сценарій його отримання) для інших систем;
- ретранслятора процедур взаємодії інших СППР (без впливу на сценарій прийняття рішень);
- транслятора протоколів у стеку протоколів СППР.

В [1, 2] розглядаються пропозиції щодо патернів (узагальнених шаблонів) взаємодії розподілених СППР. Осмислення типових способів взаємодії множин СППР дозволить розглядати їх як *відкриті системи* (незалежні від реалізації, зараз вони всі закриті) з переходом до віртуальних СППР, власне, СППР в хмарах. Тобто потрібно добитися, щоб важливою була не архітектура, а способи взаємодії, підкріплені профілем протоколів для інтерфейсів, форматів рішень та сценаріїв. Пропонуються патерни взаємодії СППР ділити на *логічні та архітектурні*.

Логічні патерни представляють розуміння з боку ОПР взаємодій розподілених СППР і складаються із таких чотирьох: 1) *довільні з'єднання*, 2) *mesh-з'єднання* (взаємодії лише із "сусідньою" СППР в термінах певної метрики [2]), 3) *родинні з'єднання* (з однотипними протоколами, архітектурами тощо) та 4) *контекстний пошук*.

Архітектурні патерни представляють реалізацію взаємодії (на базі логічних) і складаються із 2-х груп шаблонів: а) які не змінюють протоколи взаємодії та б) які змінюють протоколи. Загалом виділені наступні архітектурні патерни: 1) *безпосередньої взаємодії* (однорангової (довільне з'єднання та mesh-з'єднання (в свою чергу просте та лавиноподібне) та ієрархічної), 2) *ретранслятора* (дейтаграмний (простий, лавиноподібний) та з віртуальними каналами), 3) *ілюз* (із однієї СППР чи стеку протоколів).

Наявність, як норми організації СППР, бази рішень в СППР (локальній, спільної для розподілених СППР, СППР в хмарах) має проблему дій ОПР за стереотипом (інерцією) та проявлення домінанти (стереотипу розробників) в роботі СППР [3]. Зокрема, мала база рішень зразу приречена до дій ОПР за інерцією, тому, як кажуть, "розмір має значення...". Звичайно ж, ця проблема проявиться дещо пізніше.

Зараз же, на думку автора, головне: опанувати формальну *обробку, порівняння та накопичення* рішень як самостійних сутностей. Все інше буде лише справою техніки.

Список літератури

1. Бондарчук Ю.В. Патерни взаємодії СППР як крок до віртуалізації СППР //International conference "Problems of decision making under uncertainties" (PDMU-2012). 23-27.4.2012, Мукачево. - С. 60-61.
2. Бондарчук Ю.В. Метричні характеристики розподілених СППР //International conference "Problems of decision making under uncertainties" (PDMU-2013). 13-17.5.2013, Східниця, - С. 93.
3. Бондарчук Ю.В. Щодо проблеми виявлення інерційності СППР та ОПР у прийнятті рішень. International Workshop "Problems of decision making under uncertainties" (PDMU-2009). 5-9.10.2009, Кам'янець-Подільський. – С. 40-41.