

## ПОБУДОВА БАЗ ЗНАТЬ ДЛЯ СУМІСНОГО СТВОРЕННЯ ПРОГРАМ І ОБЛАДНАННЯ ТА ЇХ ФОРМАЛЬНОЇ ВЕРИФІКАЦІЇ

Розглянуто підхід до комплексного розв'язання задач аналітичної підготовки аналізу і синтезу програмного забезпечення. Ключові елементи такого підходу складають формати подання заголовків специфікацій і змісту ресурсів розв'язання задач (РРЗ) і база знань, що систематизує пошук аналогів, вибір близьких компонентів, використання правил перетворень та механізми декомпозиції РРЗ. Показано, що застосування таких баз знань дозволяє максимально використати попередні розробки і гранично наблизити значення цільових семантичних змінних до вимог обмежень і критеріїв щодо характеристик обробки.

The approach to the complex task decision for analytical support of software analysis and synthesis is considered. Key elements of such approach use representation formats for specification headers contents of task decision resources (TDR) and knowledge base for analogues search, a choice of best components, use of transformation rules and decomposition mechanisms for TDR. It is shown, that application of such knowledge bases allows using the most of the previous software, and making values of target semantic variables extremely close to restriction requirements and criteria of processing speed.

**Вступ.** Традиційна побудова систем автоматизованого проектування (САП) програмних і апаратних ресурсів розв'язання задач (РРЗ) спирається на табличну організацію [1] вбудованих структур даних. Більшість цих таблиць, що настроюються при реалізації мови та трансляції з вхідних мов зберігають актуальність і в сучасних засобах оптимізації, синтезу та верифікації [3, 6, 7]. Перспективним підходом до комп'ютерної автоматизації сумісного проектування програм і апаратури прискорення обробки є включення баз знань (БЗ) аналітичного супроводження рішень [2] з безпосереднім використанням робочого середовища табличних і кубічних сховищ реляційних СУБД до складу САП РРЗ.

Фактично всі правила і механізми обробки можна задати комплексами відношень, накопиченими в таблицях і кубах. Головне призначення таблиць і правил БЗ в САП РРЗ – забезпечити підготовку проектних рішень на базі аналізу наявних РРЗ і вимог до розробок [1, 3]. Інформаційно-аналітичну базу (ІАБ) САП РРЗ складають комплекси табличних і кубічних сховищ для: **граматичного аналізу** об'єктів, описів і реалізацій проблемних галузей досліджень (ПГД) і РРЗ; **визначення базових характеристик** якості РРЗ; **збереження** попередньо розроблених РРЗ разом з характеристиками їх виконання; **збереження даних про верифікацію** РРЗ, дії з верифікації та осіб, відповідальних за її виконання; **розпізнавання** проблемних областей і РРЗ за їх специфікаціями, реалізаціями та іменуваннями; **вибір**

РРЗ, що найкраще відповідають заданим критеріям.

Головна проблема використання реляційних СУБД полягає у складності обробки правил, декларацій виразів і операторів комп'ютерних мов. Одним з шляхів підвищення рівня автоматизації проектування в САП РРЗ є використання полів даних спеціального аналітичного типу [4]. Примірники таких даних визначають елементи програмних модулів і їх специфікацій з накопиченням бажаних і фактичних характеристик модулів програм і моделей апаратури. В різних САП РРЗ підготовка і реалізація проектних рішень стосуються задач аналізу і синтезу об'єктів ПГД, тощо.

**Визначення сховищ реляційної БД для задач вхідного синтаксичного і лексичного аналізу.** Лексичний аналіз (ЛА) та синтаксичний аналіз (СА) використовують ІАБ з сукупністю таблиць і списків лексем, шаблонів, правил та кубів підсумкових даних. Основні сховища даних для граматичної обробки вхідних мов включають **інформаційні таблиці**  $r_{sk}$ ,  $r_{tk}$ ,  $r_{rk}$ ,  $r_{kk}$ ,  $r_{dk}$ ,  $r_{lk}$ ,  $r_{mk}$ , **координатні таблиці**  $a_{dk}$ ,  $a_{sk}$ ,  $a_{tk}$ ,  $a_{ek}$ ,  $a_{rk}$ ,  $a_{lk}$ ,  $a_{ck}$ ,  $a_{fk}(x_{k1}, \dots, x_{ki_{\max}}, \xi_{k1}, \dots, \xi_{kj_{\max}}, a_{k1}, \dots, a_{ki_{\max}})$  та **куби**  $\prod a_{fk}(\sigma_{k1}(\xi_{k1}), \dots, \sigma_{kj}(\xi_{kj_{\max}})) | x_{k1}, \dots, x_{ki_{\max}}, \xi_{k1}, \dots, \xi_{kj_{\max}}$ , де  $k$  – номери координат;  $x_{k1}, \dots, x_{ki_{\max}}$  – поля ідентифікації;  $\xi_{k1}, \dots, \xi_{kj_{\max}}$  – поля семантично істотних даних;  $a_{k1}, \dots, a_{ki_{\max}}$  – поля додаткових даних обліку. Ключова частина структур основних таблиць та кубів підсумків граматичного аналізу показана на Рис. 1. Фактично вона охоплює всі використані таблиці граматичного аналізу і

**ПОБУДОВА БАЗ ЗНАТЬ ДЛЯ СУМІСНОГО СТВОРЕННЯ ПРОГРАМ І ОБЛАДНАННЯ ТА ЇХ ФОРМАЛЬНОЇ ВЕРИФІКАЦІЇ**

при наявності таблиць  $r_{tk}, r_{rk}, r_{kk}, r_{lk}, r_{mk}, a_{dk}, a_{tk}, a_{ek}, a_{rk}, a_{ck}, a_{vk}$  для кожної мови, реалізованої в САП РРЗ, забезпечує обробку кодів, складе-

них на різних мовах програмування і моделювання апаратури, перетворених в єдину форму внутрішнього подання.



**Рис. 1. Узагальнена інформаційно-логічна структура БЗ перетворень парсера САП**

Визначення структури сховищ БЗ внутрішніх даних семантичної обробки задач. Семантична обробка використовує ІАБ, показану на Рис. 2, з сукупністю таблиць  $r_{cm}, r_{sm},$

$r_{lm}, r_{pm}, r_{dm}, r_{lrm}, r_{lm}, r_{fm}$ , що включають специфікації даних, цілей, шаблонів, правил і шляхів розв'язання користувача і стандарти вбудованої семантичної обробки мов САП та

РРЗ. Кубічне сховище використовується для накопичення оцінок ефективності і статистик виконання окремих РРЗ, спрямованих виключно на розв'язання внутрішніх задач ПГД. Координатні таблиці куба ПГД  $a_{lm}$ ,  $a_{pm}$ ,  $a_{sm}$ ,  $a_{lm}$ ,  $a_{cm}$ ,  $a_{vm}$ ,  $a_{dm}$ ,  $a_{dvm}$  визначають обробку за-

дач підтримки прийняття рішень, якщо такі розв'язуються в рамках цієї ж ПГД.

Статистики і підсумки, накопичені в кубі ПГД, використовуються на етапі проектування для пошуку найкращої реалізації всіх РРЗ.

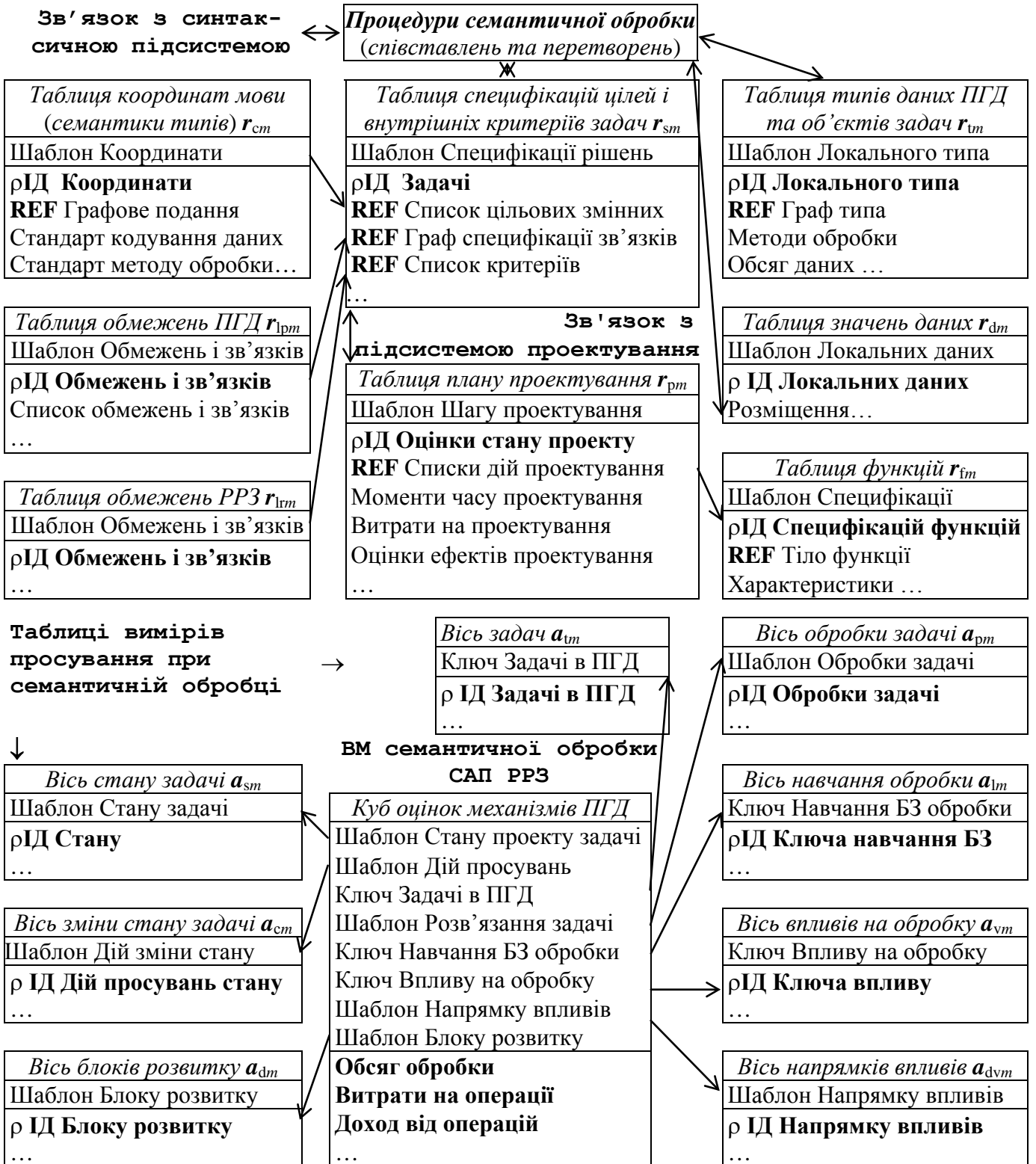


Рис. 2. Узагальнена інформаційно-логічна структура БЗ семантичних перетворень САП РРЗ

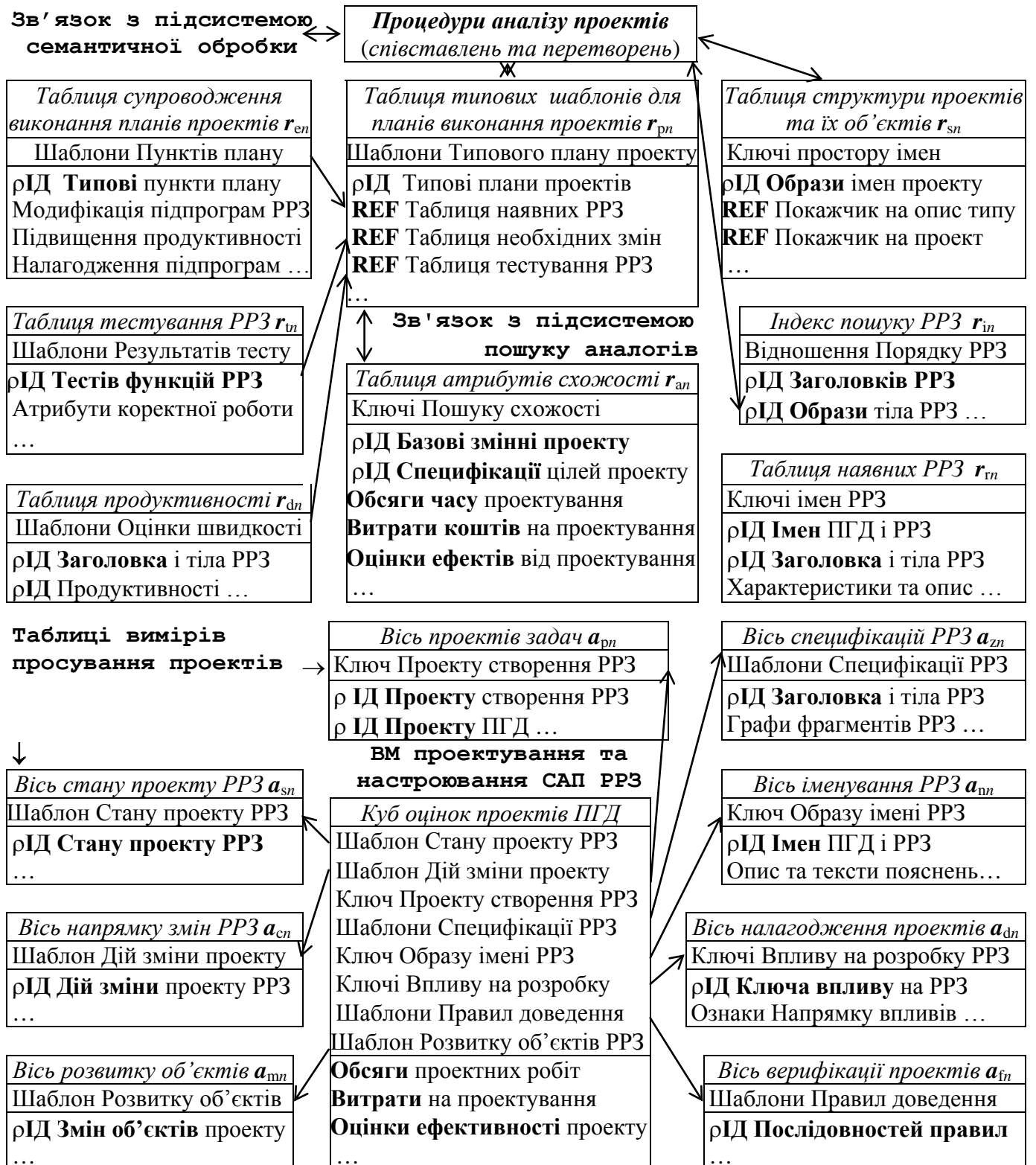
Визначення структури сховищ БЗ даних процесу проектування РРЗ. За аналогією з етапами аналізу моделей програм побудуємо

схему ІАБ для процесу проектування. Проектування РРЗ використовує ІАБ, показано на Рис. 3, з сукупністю таблиць  $r_{cm}$ ,  $r_{pm}$ ,  $r_{sm}$ ,  $r_{tm}$ ,  $r_{lm}$ ,

**ПОБУДОВА БАЗ ЗНАТЬ ДЛЯ СУМІСНОГО СТВОРЕННЯ ПРОГРАМ І ОБЛАДНАННЯ ТА ЇХ ФОРМАЛЬНОЇ ВЕРИФІКАЦІЇ**

$r_{an}$ ,  $r_{dn}$ ,  $r_{in}$ , що зберігають дані про проекти РРЗ, плани і стан їх реалізації, шаблонів, правил та РРЗ спроектованих і в процесі проектування. Кубічне сховище використовується для накопичення оцінок стану створення про-

ектів ПГД і окремих РРЗ. Координатні таблиці куба проектів  $a_{pn}$ ,  $a_{zn}$ ,  $a_{sn}$ ,  $a_{nn}$ ,  $a_{cn}$ ,  $a_{dn}$ ,  $a_{mn}$ ,  $a_{fn}$  задають напрямки підтримки прийняття рішень при реалізації проектів в одній ПГД.



**Рис. 3. Узагальнена інформаційно-логічна структура БЗ процесів проектування САП РРЗ**

Оскільки весь процес створення РРЗ пов'язаний зі способами подання специфікацій, про-

грам, моделей і процесів доведення [7] важливо використати узагальнений механізм формуван-

ня аналітичних моделей функціональності обчислень і всіх можливих оцінок їх складності. Такий механізм створюється на базі узагальненого абстрактного типу аналітичних даних [4], в якому використовуються спеціальні канонічні форми зберігання [4, 6] для полегшення зіставлень варіантів рішень програм і доведень.

Вимоги до канонічних форм подання кодів програм та протоколів верифікації [5] спираються на відношення лінійного порядку  $\leq$  [6] на базі кодів типів різних термінальних і нетермінальних вузлів УСГ. Термінальні вузли включають поля образів ідентифікації змінних або внутрішню форму подання константних даних:

- **впорядковуюча нумерація кодів типів** термінальних та нетермінальних вузлів УСГ;
- **впорядкування або нумерація змінних** через запам'ятовування імен або ідентифікаторів, заданих за будь-якою системою довільних евристичних правил [6];
- **впорядкування константних даних** за їх типами і значеннями у внутрішній формі;
- **можливість встановлення відношення канонічного лінійного порядку** для комутативних операцій серед виразів та операторів;
- **можливість побудови формул різних оцінок складності** для визначення точності, надійності та ефективності варіантів РРЗ;
- **можливість підрахунку значень різних оцінок складності** для конкретизованих варіантів РРЗ;
- **можливість порівняння аналітичних і числових оцінок складності** для вибору кращого варіанта реалізації РРЗ.

Додатково розглянемо можливості створення РРЗ для конструктивних задач. Звичайно вони в технічних ПГД вони обмежуються розрахунковими задачами аналізу і синтезу з наступною перевіркою відповідності розрахунків гранично припустимим значенням. В задачах аналізу обробляються дані про навантаження конкретизованого технічного об'єкта на рівні його моделі. В задачах синтезу визначаються параметри елементів технічних об'єктів на основі граничних значень їх навантажень. При розв'язанні базових класів конструктивних задач використовують наступні процедури розв'язання часткових задач та задач проблемного рівня САП РРЗ:

- **часткове підбиття підсумків в задачах аналізу поточного стану та рішень** з використанням технологій накопичення та вибірки підсумків в гіперкубах [2];

- **оцінка навантажень на елементи об'єктів**, що виконується шляхом прямих обчислень або розв'язання рівнянь операціями перетворень даних аналітичного типу;
- **підготовка рішень та формування значень параметрів елементів об'єктів** розрахунковими формулами, одержаними через зворотні перетворення аналітичних даних;
- **оцінки обмежень навантажень і параметрів об'єктів** для оцінки працездатності, надійності і ефективності проектних рішень.

Таким чином, показані 4 класи задач створюють основу реалізації комплексу РРЗ для будь-якої ПГД, в тому числі і для побудови САП. Для задач оцінок варіантів проектів визначають гіперкуби  $\Pi r_{ik}$ , таблиці  $r_{ck}$  та прості або підсумкові змінні оцінок  $\xi_j$  та  $\sigma_{kj}(\xi_{kj_{max}})$ , які входять до критеріїв рішень ПГД.

**Умови коректного розв'язання задач САП зі створення надійних і ефективних РРЗ.** Важливо, щоб БЗ дозволила знайти найкращі РРЗ, що розв'язують близькі задачі, і автоматизованим шляхом побудувати необхідний залишок РРЗ, що проектується. Контроль точності, надійності, ефективності та своєчасності розв'язання забезпечується специфікаціями РРЗ вторинного рівня.

Доказові перетворення виразів специфікацій зв'язків виконуються вбудованими операціями аналітичного типу даних з контролем вторинних цілей і критеріїв, заданих специфікаціями ПГД і РРЗ [4]. Для перетворень важливо визначити шлях формування прирощення тексту коду  $P_{pn}$  при побудові модифікованого РРЗ і оцінки прирощень потрібних елементів критеріїв. Спочатку треба визначити різницю між специфікацією  $S_{pn}$  і  $P_{pn}$ , а потім трансформувати її в код додатку до базового коду РРЗ для формування результату  $\Delta P_{pn} = f(y, S_{pn} - P_{pn})$ . При відсутності базового коду перетворення синтезу формуються як послідовність прирощень, що формуються з первинної специфікації з врахуванням заданих критеріїв.

**Визначення загальних правил проектування РРЗ та настроювання роботи САП.** З одного боку в шлях просування при синтезі кодів визначається або вручну, або автоматизованою підказкою прирощення. Якщо при автоматизованому синтезі необхідно зробити, що за попереднім доведенням відповідає специфікації, то при ручному синтезі необхідно формально довести коректність перетворення.

З іншого боку треба автоматизувати генерацію різних варіантів повний перебір яких є NP-

складною задачею [6]. Однак перший ліпший варіант програми можна побудувати через перетворення, що послідовно наближують створення повного коду РРЗ, а потім удосконалювати його результатами адекватних перетворень, які покращують значення критеріїв.

Кількість можливих шляхів  $n_c$  перетворення визначається кількістю  $n_k$  відомих методів потенційно точного або наближеного розв'язання задач, занесених до БЗ САП РРЗ. Ці кількісні оцінки визначаються кількістю і розмірністю вхідних  $n_i(N_{ik})$ , вихідних  $n_o(N_{ok})$  та робочих змінних  $n_w(N_{wk})$ , сформованих аналітичними методами підготовки розв'язання задач

$$n_k = n_k(n_i(N_{ik}), n_w(N_{wk}), n_o(N_{ok})). \quad (1)$$

Як правило, кількість ефективних варіантів розв'язання найбільш досліджених задач не перевищує десятків, а для одержання технічно ефективного результату можна обмежитися першою десяткою методів.

**Вибір порядку створення задач та підзадач в САП РРЗ.** Звичайно після первинного визначення типів даних (координат) будь-якого супероб'єкта ПГД доцільно створити набір типових комплексів задач для розв'язання в цій галузі. Розмірність комплексів задач визначається кількістю та розмірністю характеристик конкретизацій об'єктів, включених до моделі конкретної комплексної ситуації в ПГД. Кількість задач  $n_d$  обмежується кількістю об'єктів та агрегатів даних, які можуть бути вхідними та вихідними в підпрограмах. Кількість задач буде істотно менше кількості попарних комбінацій об'єктів, як вхідних та вихідних даних. Таким чином,  $n_d$  залежить від кількості та структурованості даних в об'єкти для їх використання в задачах як вхідних і вихідних даних.

Таким чином, для доказової розробки, верифікації і оперативного настроювання РРЗ вдосконалені САП повинні виконувати наступну послідовність дій, пов'язаних з вмістом вбудованої БЗ супроводження проектів та аналітичних перетворень.

1. Визначити базову структуру типів даних (координат) ПГД та комплексів об'єктів для розв'язання задач аналізу та синтезу.

2. Визначити цільові змінні та вторинні критерії в специфікаціях окремих задач ПГД.

3. Організувати адекватні перетворення за правилами перетворень аналітичних даних і методами розв'язання задач, занесеними до БЗ семантичних перетворень, показаної на рис.2.

4. Організувати пошук аналогів та супроводження реалізації проектів з використанням БЗ процесів проектування, показаної на рис. 3.

Процес доказової підготовки та настроювання РРЗ повинен супроводжуватись аналітичним контролем функціональності і продуктивності модулів, критичних за часом виконання [4].

### **Висновки**

Таким чином, для підвищення рівня автоматизації САП РРЗ доцільно істотно розширити вбудовану БЗ до блоків управління обробкою, а також проміжні підсумки можуть використовуватися для ефективної комбінації фрагментів кодів. Ключовим елементом організації таких БЗ є використання абстрактного типу аналітичних даних. Кількість задач аналізу і синтезу в конкретному комплексі ПГД істотно обмежена кількістю істотних груп об'єктів та їх характеристик. Накопичення даних про ефективні реалізації РРЗ в БЗ є спеціальним варіантом задачі вибору ефективних рішень за допомогою БЗ, реалізованих в рамках реляційних СУБД [2].

### **Список посилань**

1. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии, инструменты: Пер. с англ. – М.: Издательский дом Вильямс, 2001. – 768 с.
2. Бэлсон Д., Гокмен М. Ингрэм Дж. Внутренний мир Oracle8. Проектирование и настройка: – К.: Изд-во "ДиаСофт". – 2000. – 800 с.
3. Лисков Б., Гатэг Дж. Использование абстракций и спецификаций при разработке программ. Пер. с англ. М.: Мир, 1989. 424 с.
4. Пустоваров В.І. Супроводження контролю відповідності програм і моделей формальним специфікаціям задач // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка – К.: «Век+». 2007, 47, с. 269-279.
5. Hehner E.C.R. Practical theory of programming. Springer-Verlag, New York, 1993 – 243 p.
6. Metzger R.C., Zhaofang W. Automatic algorithm recognition and replacement: a new approach to program optimization / The MIT Press, Cambridge, 2000. 219 p.
7. Woodcock J., Davies J. Using Z. Specification, Refinement, and Proof. C.A.R. Hoare Series editor, 1995 – 390 p.