

УДК 004.652.43

**С. О. Кулик***Українська академія друкарства***МОДЕЛІ УНІТЕРМІВ ТА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ  
ОПISУ ЗАВДАНЬ НА СТВОРЕННЯ БАЗ ДАНИХ**

*Розглядається модель типізованого шляху як засіб опису об'єктів реляційної моделі. Побудовано математичну модель функційної частини інформаційної технології, яка описує створення і використання системи шаблонів у процесах синтезу доменів, заголовків відношень, значень відношень, збереження у бібліотеках шаблонів і синтезованих формул завдання на генерування та підвищує рівень автоматизації процесів синтезу завдань на створення реляційних баз даних.*

***Алгебра алгоритмів, модель, унітерм, інформаційні технології***

Найбільш відомі і часто використовувані інформаційні технології, наприклад, Microsoft Visual Studio .NET [4], для створення реляційних баз даних використовують модель візуального-текстового проектування. Елементами інтерфейсу користувача є візуальні дерева, діаграми, табличні подання, текстові редактори.

Відомі інформаційні технології, хоча і комп'ютеризують завдання на створення реляційних баз даних, але не подають у формалізованому вигляді. У зв'язку з цим, до таких завдань не можуть застосовуватися математичні методи, наприклад, для оптимізації завдань за кількістю унітермів-складових. Застосування графічно-текстової моделі до баз даних є трудомістким процесом та потребує великих затрат часу. Модель даних має особливе значення в інформаційних системах. Дані, що зберігаються з першого дня впровадження системи, мають надзвичайну вартість. Однак у процесі експлуатації системи виникає потреба у модифікації моделі або розширення її семантики. У разі, якщо при проектуванні моделі даних були допущені помилки, модифікація або розширення моделі потребуватиме перебудови, доопрацювання бази даних та програмного коду інтерфейсу до даних. Такі зміни можуть мати вартість, яка перевищує вартість системи або призвести до пошкодження, або втрати нагромадженої у системі інформації. Отож проектування моделі даних потребує дотримання правил проектування реляційних баз даних, які закладені у теорії нормалізації. Такими є вимоги до атомарності значень компонентів кортежів, відсутність надлишковості у змісті відношень (аномалії додавання, видалення, оновлення). Дослідження на відповідність даних означеним вимогам виконується на основі нормальних форм відношень, які були означені дослідниками Рэймондом Бойсом та Едгаром Коддом. Теорія нормалізації ґрунтується на теорії множин, поняттях проекції та функційної залежності даних один від одного. Наявний засіб створення баз даних, мова запитів SQL не є формалізованими. Таким чином не існує опису, який міг би поєднати математичні засоби дослідження моделі даних із завданням на утворення

моделі даних. Алгебра алгоритмів є формалізованою та має такі засоби. За допомогою функційних унітермів алгебри алгоритмів можливо створювати аналітичні формули, які реалізують певні етапи дослідження даних, наприклад, перевірку атомарності, пошук надлишковості, аналіз функційної залежності. На основі простих функційних унітермів можливе створення комплексних засобів аналізу — функційних унітермів, які реалізують аналіз набору відношень загалом. До таких функційних унітермів можна передавати відношення як параметри та отримувати результати у вигляді скоректованих відношень, текстових результатів або рекомендацій для поліпшення проаналізованої частини моделі.

Для розв'язання завдань створення та аналізу моделі даних граматика мови опису завдань синтезу баз даних має оперувати всіма об'єктами реляційної моделі. Розв'язання цього завдання потребує означення засобів доступу у семантиці мови.

Синтаксис і семантика мови інформаційної технології опису завдань на створення реляційних баз даних використовує операції алгебри алгоритмів [4]. Семантика мови визначається за допомогою унітермів [6]. Унітерм доступу має важливе значення для граматики мови та дає основу для створення унітермів, які виконують дії над об'єктами реляційної моделі.

Відомими у комп'ютерних науках описами доступу є веб-адреси URL (Uniform Resource Locator), шляхи до файлів в операційних системах, оператор доступу у мовах програмування. Модель доступу до об'єктів реляційної моделі має відповідати вимогам зрозумілості для людини та однозначності. Частина мови, яка описує семантику доступу до об'єктів, є синтаксисом і семантикою шляху.

Модель типізованого шляху. *Означення 1:* Типізований шлях (або скорочено шлях) є виразом який однозначно вказує на об'єкт реляційної моделі або його властивість та забезпечує доступ до будь-якого об'єкта цієї моделі у будь-якій частині формули будь-якої частини опису.

У моделі типізованого шляху, вказуються не тільки назви, але і типи складових частин. Такий підхід дає можливість скорочувати кількість складових частин залежно від загального контексту формули.

*Примітка:* мова оперує також об'єктами, які не є елементами реляційної моделі. Такими є методи, змінні, файли та іменовані масиви, — вони мають схожий синтаксис, але семантика шляху не включає їх опис.

Типізований шлях складається із секвенції адрес  $s_q$ ,

$$\rho = \overbrace{s_0 : s_1 : \dots : s_{m-2} : s_{m-1}} \quad (1)$$

де  $s_0, s_1, s_{m-2}, s_{m-1} \in s_q$  — адреси шляху;  $q$  — позиція адреси у шляху;  $m$  — число елементів шляху, визначається глибиною вкладеності цільового об'єкта, який є за адресою  $s_{m-1}$ .

Одна адреса вказує тільки на один об'єкт певного типу та дає доступ наступним адресам у шляху до властивостей або дочірніх об'єктів, які йому належать.

У шляху може бути одна або декілька адрес. Адреса  $s_{i-1}$ , яка стоїть зліва від даної адреси  $s_i$  є батьківською, та навпаки адреса  $s_{i+1}$ , що є справа від  $s_i$  називають дочірньою. Тобто, адреси впорядковані зліва направо, відповідно позиція адреси у шляху, робить його старшим відносно адрес справа від нього та молодшим відносно адрес зліва від нього.

Означення 2. Адреса  $s_q$  залежно від позиції приймає значення:

$$s_q = \overbrace{\delta_q; \dots; \delta_q; (q=0)-?}, \quad (2)$$

де крапкою позначено унітерм доступу до полів батьківського елемента;  $\delta_q$  — змінна адреси, яка описується виразом:

$$\delta_q = \overbrace{\tau_q; [\dots; \varepsilon_q; \dots]; \tau_q; (\tau_q \in Q_{r0})-?}, \quad (3)$$

де  $\tau_q$  — вказівник типу об'єкта;  $\varepsilon_q$  — ідентифікатор об'єкта, назва за якою до нього можна звертатися та відрізнити від інших об'єктів того самого типу.

Синтаксис змінної адреси обов'язково потребує використання квадратних дужок, у які береться ідентифікатор.

Область значень вказівника типу об'єкта ( $\tau$ ) визначена формулою:

$$\tau_q \in \overbrace{Q_{r0}; Q_{r1}; Q_{r2}}, \quad (4)$$

де  $Q_{r0}$  — секвентна область іменованих об'єктів. Іменовані об'єкти — це об'єкти реляційної моделі баз даних, до яких можна звертатись за ідентифікатором;  $Q_{r1}$  — секвентна область неіменованих об'єктів, об'єкти реляційної моделі, які не мають ідентифікатора, але обов'язково мають іменованій батьківський елемент, за допомогою адреси якого він є доступним у шляху;  $Q_{r2}$  — секвентна область властивостей об'єктів. Кожен об'єкт реляційної моделі може мати одну або декілька властивостей. Кожна властивість обов'язково має значення, яке можна отримати або завдати. Сукупність властивостей обумовлюють поведінку об'єкта. набір властивостей є постійним та загальним для певного типу об'єктів системи.

$$Q_{r0} = \overbrace{DB, R, A, t; D, q, Ix, W; Fk, Pk, \alpha, \omega, \mu}, \quad (5)$$

де  $DB$  — ідентифікатор бази даних;  $R$  — відношення;  $A$  — атрибут відношення;  $t$  — секвентний набір значень відношення;  $D$  — домен бази даних [1];  $q$  — запит до відношень;  $Ix$  — індекс;  $W$  — зображення відношення;  $Fk$  — первинний ключ відношення;  $Pk$  — вторинний ключ відношення [2];

$$Q_{r1} = \overbrace{c, b, h, k} \quad (6)$$

де  $c$  — колекція обмежень відношення;  $b$  — зміст відношення;  $h$  — заголовок відношення;  $k$  — елемент секвентного набору значень (компонент).

$$Q_{r2} = \overbrace{\left( \begin{array}{c} ID \\ a0 \\ path \\ Engine \end{array} ; \begin{array}{c} KeyA \\ RefR \\ RefA \\ action \end{array} ; \begin{array}{c} defaultvalue \\ alias \\ TYPE, a1 \\ CAPACITY, a2 \end{array} ; \begin{array}{c} PREDICATE, a3 \\ NOTNULL, a5 \\ AUTOINCREMENT \\ a6 \end{array} ; \begin{array}{c} errormessage \\ outputformat \\ inputmask \\ v \end{array} \right)} \quad (7)$$

де  $ID$  — властивість доступу до ідентифікатора об'єкта, є загальним для об'єктів усіх типів із  $Q_{r0}$ ;  $a0$  — псевдонім  $ID$  для домену  $D[\varepsilon]$ ;  $path$  — шлях до бази даних  $DB[\varepsilon]$ ;  $Engine$  — тип ядра бази даних  $DB[\varepsilon]$ ;  $KeyA$  — ключовий атрибут зовнішнього ключа  $Fk[\varepsilon]$  або первинного ключа  $Pk[\varepsilon]$  у обмеженні відношення  $R[\varepsilon].c$ ;  $RefR$  — батьківське відношення — словник зовнішнього ключа  $Fk[\varepsilon]$  у обмеженні відношення  $R[\varepsilon].c$ ;  $RefA$  — ключовий атрибут батьківського відношення у обмеженні відношення;  $action$  — дія при зміні первинного ключа  $Pk[\varepsilon]$  у обмеженні відношення  $R[\varepsilon].c$ ;  $defaultvalue$  — значення за замовчуванням атрибута  $A[\varepsilon]$ ;  $alias$  — псевдонім атрибута  $A[\varepsilon]$ , за яким його відображено у інтерфейсі користувача системи;  $TYPE = a1$  — тип домена  $D[\varepsilon]$ ;  $CAPACITY = a2$  — розмір домена  $D[\varepsilon]$ ;  $PREDICATE = a3$  — предикат домена  $D[\varepsilon]$ ;  $NOTNULL = a5$  — логічне значення, яке дозволяє або забороняє домену  $D[\varepsilon]$  приймати порожнє значення —  $null$ ;  $AUTOINCREMENT = a6$  — логічне значення, яке задає або відмінює автоприрощення значення домену  $D[\varepsilon]$ ;  $index$  — логічне значення, яке задає або відмінює обмеження на повторювання значень у компонентах  $k$  атрибута  $A[\varepsilon]$ ;  $errormessage$  — текст повідомлення, яке має виводитися на екран у разі порушення обмежень, які накладено на атрибут  $A[\varepsilon]$  доменом  $D[\varepsilon]$ ;  $inputmask$  — маска вводу значень до атрибута  $A[\varepsilon]$ ;  $outputformat$  — формат виводу значень для атрибута, який має складові типи даних, наприклад, такі як фінансові, дати/часу та інші;  $v$  — значення компонента  $k$ .

Для усіх властивостей домену, заради зручності використання у формулах, мовою передбачені скорочені псевдоніми у вигляді  $a0 - a3, a5 - a6$ .

Назва об'єкта ( $\varepsilon$ ) описується формулою:

$$\varepsilon = \overbrace{\varphi_0 ; \varphi_1 ; \dots ; \varphi_{n-2} ; \varphi_{n-1}} \quad (8)$$

де  $\varphi_0, \varphi_1, \varphi_{n-2}, \varphi_{n-1}$  — символи Унікоду, обмежені секвентними областями допустимих значень.

Унітерми секвентної області у формулі (8) мають відповідати певним умовам:

$$\varphi_0 = \overbrace{Q'_0 : Q'_1}, \tag{9}$$

де  $\varphi_0$  — перший символ назви;

$$\varphi_1, \dots, \varphi_{n-2}, \varphi_{n-1} \in Q_1'', \tag{10}$$

$\varphi_1, \dots, \varphi_{n-2}, \varphi_{n-1}$  — решта символів назви.

Для (9) та (10) області значень описуються формулою

$$Q_1'' = \overbrace{\overbrace{Q'_0 : Q'_1} : Q'_2}, \tag{11}$$

де  $Q'_i$  — знак підкреслення ( $Q'_i = \_$ );

$$Q'_0 = \overbrace{\overbrace{A : B} : \dots : Z} : \overbrace{a : b} : \dots : z}, \tag{12}$$

де  $Q'_0$  — літери латинського алфавіту,

$$Q'_2 = \overbrace{0 : 1} : \dots : 9 \tag{13}$$

$Q'_2$  — десяткові цифри;

Належність  $\varphi_0$  до секвентних областей  $Q'_0$  та  $Q'_1$  означає, що ідентифікатор має починатися тільки з літери латинського алфавіту або знака підкреслення.

Здатність ідентифікувати об'єкти ґрунтується на властивості унікальності назви для заданого типу об'єктів.

Якщо область ідентифікаторів означеного типу  $\tau_n$  позначити  $Q\tau_n$ , то

$$Q\tau_n = \overbrace{\varepsilon_0 : \varepsilon_1} : \dots : \varepsilon_{m-2} : \varepsilon_{m-1}}, \tag{14}$$

де,  $m$  — кількість елементів секвентної області  $Q\tau_n$ , а для кожного  $\varepsilon_k \in Q\tau_n$  використовується така формула:

$$r = \left( \begin{array}{l} \text{if } (i \in \text{int}) = 0 \\ \overbrace{i++ : (\uparrow \text{true} : (\varepsilon_k \neq \varepsilon_i) - ?) : c_i : (i < m) - ?} \\ \vdots \\ c_i \\ \uparrow \text{false} \\ \vdots \\ \overbrace{(\varepsilon_k \notin Q\tau_n : \varepsilon_k \in Q\tau_n : (r = \text{true}) - ?} \\ \vdots \\ Q\tau_n = \varepsilon_k \end{array} \right) \tag{15}$$

де  $i = 0 \dots m-1$  — номер елемента в області;  $Q\tau_n$ ;  $k \in 0 \dots m-1$ ;  $c_i$  — стандартний унітерм алгебри алгоритмів, який описує повернення у цикл за змінною (умовою)  $i$ ;  $r$  — результат який повертається формулою циклічного елімінування перевірки ідентифікатора об'єкта  $\varepsilon_k$  на унікальність відносно наявних

елементів  $\varepsilon_i$  секвентної області  $Q\tau_n$ .  $\uparrow$  — унітерм повертання значення на вхід операції агрегування.

*Означення 3:* Якщо хоч один адрес, який складає секвенцію шляху не існує за приведеним ідентифікатором  $\varepsilon$  або об'єкти даного типу не є дочірніми для батьківського адресу, то такий шлях повертає порожній унітерм (\*).

*Примітка:* Наведена модель ідентифікатора  $\varepsilon$ , яка використовуватиме для шляху, є універсальною для усіх інших іменованих об'єктів мови таких як метод  $\mu$ , змінна  $\omega$ , асоціативний масив  $\lambda$ .

*Модель значення.* Властивості об'єктів набувають певних значень. Наприклад,  $\sigma$  — вираз значення, може бути числовим, символьним, логічним або іншим значенням із типів, які означені стандартом ANSI SQL-92 для сучасних серверів баз даних.

Загальна формула опису значення  $\sigma$  описується виразом (16):

$$\sigma = \overbrace{"; \theta ; "}, \quad (16)$$

де  $\theta$  — змінна, значення якої описується формулою

$$\theta = \overbrace{\overbrace{\psi_0 ; \psi_1 ; \dots ; \psi_{g-2} ; \psi_{g-1}},} \quad (17)$$

де  $g$  — кількість символів у значенні, яка називається довжиною;  $\psi_0, \psi_1, \psi_{g-2}, \psi_{g-1}$  — символи Унікоду які обмежені секвентною областю значень  $Q_0$ ».

$$\psi_m \in Q_0'' = \overbrace{\overbrace{\overbrace{Q_0', Q_1', Q_2', Q_3', Q_4', Q_5', Q_6'},} \quad (18)$$

де  $m = 0..g-1$  — позиція символу у значенні змінної  $\theta$ ;  $Q_0'$  — літери латиниці (12);  $Q_1'$  — знак підкреслення (11);  $Q_2'$  — десяткові цифри (13);  $Q_3'$  — літери кирилиці.

$$Q_3' = \overbrace{\overbrace{A ; B ; \dots ; Я ; a ; б ; \dots ; я},} \quad (19)$$

де  $Q_4'$  — символи граматики;

$$Q_4' = \overbrace{\overbrace{\dots ; : ; ; ! ; ? ; " ; ' ; - ; -},} \quad (20)$$

де  $Q_5'$  — символи математичних операцій;

$$Q_5' = \overbrace{\overbrace{+ ; - ; = ; * ; / ; ^ ; \& ; ; | ; < ; ; > ;},} \quad (21)$$

де  $Q_6'$  — дужки та парні лапки;



Із формули (10) для  $\varepsilon_0 - \varphi_0 \in Q'_0$ ,  $\varphi_1, \varphi_2 \in Q'_0$ , для  $\varepsilon_1 - \varphi_0 \in Q'_0$ ,  $\varphi_1, \varphi_8 \in Q'_0$ , для  $\varepsilon_2 - \varphi_0 \in Q'_0$ ,  $\varphi_1, \varphi_2 \in Q'_0$ , що відповідає виразам формул (9), (10).

Ліва частина виразу (24) відповідає моделі шляху. Права частина виразу (24) є значенням  $\sigma$ . Значення змінної є таким

$$\theta = \overbrace{A; 9; 5}, \quad (30)$$

та розкриваючи формулу (17) маємо

$$\begin{aligned} \psi_0 &= A \in Q_0', \\ \psi_1 &= 9, \psi_2 = 5 \in Q_2'. \end{aligned} \quad (31)$$

Права частина виразу (23) відповідає секвентній області допустимих значень  $Q_0$ ».

**Модель декомпозиції функціональної підсистеми.** Синтез формул алгебри алгоритмів засобами редактора формул є текстово-графічним [6]. Створення текстових виразів унітермів є доволі трудомістким процесом для користувача та потребує певної кваліфікації від користувача. Для полегшення засвоєння синтаксису та семантики мови опису завдань на генерування баз даних є доцільною побудова автоматизованої комп'ютерної системи синтезу формул за параметрами, які задаються графічними засобами. Інтегрована у комп'ютерну систему автоматизованого синтезу формул бібліотека шаблонів дозволяє користувачу швидко засвоїти мову та багаторазово використовувати типові рішення, надає певний каркас системи, який змінюється користувачем під власну задачу. Комп'ютерна система автоматизованого синтезу формул декомпована на підсистему редактора формул @R, графічну %G та функціональну @F.

Функціональна підсистема @F інформаційної технології є рівністю

$$@F = @BDMaster,$$

де @BDMaster — підсистема, яка реалізує створення та збереження до бібліотеки формул опису складових частин реляційної моделі на основі подій, які відбуваються у елементах інтерфейсу користувача.

Модель декомпозиції функціональної підсистеми @BDMaster описується формулою (32) :

$$@BDMaster = \overline{\left[ \begin{array}{c} K_1 \\ \vdots \\ Z_1 \end{array} ; \left[ \begin{array}{c} H_1 \\ \vdots \\ H_2 \\ \vdots \\ L_1 \\ \vdots \\ C_1 \end{array} ; \left[ \begin{array}{c} A_1 \\ \vdots \\ D_1 \\ \vdots \\ T_1 \\ \vdots \\ W_1 \end{array} \right] \right]} , \quad (32)$$

де  $K_1$  — конструктор підсистеми;  $Z_1$  — глобальні змінні;  $H_1$  — функціональні унітерми опрацювання подій кнопок;  $H_2$  — функціональні унітерми опра-



цювання подій текстових полів;  $L_1$  — функціональні унітерми опрацювання подій списку формул;  $C_1$  — функціональні унітерми опрацювання подій багато текстових елементів (ComboBox);  $A_1$  — функціональні унітерми генерування об'єктів формул алгебри алгоритмів;  $D_1$  — функціональні унітерми роботи з бібліотекою шаблонів;  $T_1$  — функціональні унітерми побудови візуальних таблиць;  $W_1$  — функціональні унітерми опрацювання універсальних та глобальних подій вікна та вкладок, зокрема подій клавіатури.

Конструктор є обов'язковим унітермом підсистеми, з якого починається утворення об'єкта вікна у пам'яті облікової системи. Функціональний унітерм конструктора  $K_1$  описується формулою

$$K_1 = pbl BDMst(). \quad (33)$$

Результати обчислень, проміжні результати роботи алгоритмів, параметри, конфігурація об'єктів, налаштування процесів зберігаються протягом усього циклу існування підсистеми у пам'яті. Для виконання цієї вимоги дані мають зберігатися як глобальні змінні. Глобальні змінні  $Z_1$  у підсистемі  $@F$  описуються застосуванням операції некомутативного паралелення, унітермами, які їх визначають та ініціалізують такими значеннями:  $(TplAtbLst \in ArrLst) = new ArrLst()$  — колекція, яка зберігає назви компонентів кортежа, де  $ArrLst$  ( $ArrayList$ ) — відома підсистема, яка підтримує зберігання елементів будь-якого типу [5], забезпечує доступ до елементів за індексом, сортування та пошук елементів;  $(AtbPrpLst \in ArrLst) = new ArrLst()$  — колекція, яка зберігає назви властивостей атрибутів;  $(DmnPrpLst \in ArrLst) = new ArrLst()$  — колекція, яка зберігає назви властивостей домену;  $(DmnTpsLst \in ArrLst) = new ArrLst()$  — колекція, яка зберігає назви доменів;  $pbl sttc (DmnHist \in str[,]) = new str[22,12]$  — масив, що зберігає комірки візуальної таблиці параметрів синтезу формули доменів, де  $pbl$  ( $public$ ) — модифікатор публічного доступу до змінної,  $sttc$  ( $static$ ) — модифікатор статичного члена підсистеми [5], який надає функційному унітерму або змінній належність до  $@BDMaster$ , а не її екземплярів,  $str[,]$  — двомірний масив текстових значень;  $pbl sttc (tupleHist \in str[,]) = new str [22,12]$  — масив, який зберігає комірки візуальної таблиці параметрів синтезу формули змісту відношення (кортежів);  $pbl sttc (TblHist \in str[,]) = new str[22,12]$  — масив, який зберігає комірки візуальної таблиці параметрів синтезу формули заголовку відношення;  $pbl sttc (DmnNum \in int) = MnFrm.MasterConfig[0]$  — змінна кількості доменів, яка подаватиметься у візуальній таблиці доменів;  $pbl sttc (DmnPrpsNum \in int) = MnFrm.MasterConfig[1]$  — змінна кількості властивостей доменів, яка наводитиметься у візуальній таблиці доменів;  $pbl sttc (TblAtbNum \in int) = MnFrm.MasterConfig[2]$  — змінна кількості атрибутів у заголовку відношення, яка буде подана у візуальній таблиці відношень;  $pbl sttc (TblAtbPrpsNum \in int) = MnFrm.MasterConfig[3]$  — змінна кількості додаткових властивостей відношень, що наведена у візуальній таблиці відношень;  $pbl sttc (TplsNum \in int) = MnFrm.MasterConfig[4]$  — змінна кількості кортежів, яка буде подана у візуальній таблиці кортежів;  $pbl sttc (TplAtbNum \in int) = MnFrm.MasterConfig$

[5] — змінна кількості компонентів кортежа, що наводиться у візуальній таблиці кортежів;  $pbl (TblHndCou \in int) = 0$ ,  $pbl (DmnHndCou \in int) = 0$ ,  $pbl (TplHndCou \in int) = 0$  — лічильники кількості активізації вкладок «Відношення», «Домени», «Кортежі»;  $(fstst \in int) = 0$  — змінна, яка зберігає кількість запуску обробників подій елементів та за допомогою якої аналізується чи вперше спричинено обробник даної події;  $pbl sttc (scale \in str) = '1,5'$  — змінна — коефіцієнт масштабування ширини візуальних таблиць;  $sttc (ScrLTbx \in \%TxtBx)$  — змінна, яка є сполучним посиланням між обробником події прокрутки колеса миші та всіма текстовими полями, в яких є потреба автоматично змінювати зміст числового значення;  $pbl sttc (MstTm \in Trm) = new Unt('')$  — унітерм, який записує згенеровану системою формулу. ??? Є публічною статичною змінною, за рахунок чого надає іншим підсистемам доступ до результатів роботи системи синтезу формул опису завдання на генерування реляційної бази даних;  $sttc (mf \in MnFrm) = (MnFrm) App.Crn.MnWnd$  — змінна головного вікна підсистеми @R, де App — відома підсистема, яка задає методи взаємодії з комп'ютерною системою як з додатком операційної системи, Crn — унітерм, який вказує на наявний додаток, MnWnd — унітерм, що повертає головне вікно додатка [3].

Кнопки є найпростішими графічними елементами інтерфейсу, засобами яких користувач може кліками миші запускати виконання певних алгоритмів за рахунок опрацювання відповідної події для кожної кнопки. Функціональні унітерми опрацювання подій кнопок  $H_1$  описуються формулою:

$$H_1 = \left[ \begin{array}{l} prv (*bdel\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*badd\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnAdDscClk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnRememberBase\_Click (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnShowTab\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnDToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnRToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btntToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ (*btnGnPrvMsWhl (sdr \in obj, e \in MsWhlEvArgs)) \\ \vdots \\ prv (*GtTplClk (sdr \in obj, e \in RtdEvArgs)) \\ \vdots \\ prv (*btnTblTXML\_TTPOpn (sdr \in obj, e \in TTPEvArgs)) \\ \vdots \\ prv (*btnDmnTXML\_TTPOpn (sdr \in obj, e \in TTPEvArgs)) \\ \vdots \\ prv (*btnTplTXML\_TTPOpn (sdr \in obj, e \in TTPEvArgs)) \end{array} \right. , \quad (34)$$

де  $prv (*)bdel\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки видалення рядка у змісті візуальної таблиці, де  $prv (private)$  — модифікатор обмеженого приватного доступу до змінної або функційного унітерма, який робить функційний унітерм приватним, тобто відкритим тільки для підсистеми  $@BDMaster$ ,  $sdr \in obj$  — об'єкт, у якому відбулася подія,  $obj (object)$  відомий системний об'єкт від якого наслідуються всі інші типи та об'єкти [3, 5],  $e \in RtdEvArgs (RoutedEventArgs)$  типова відома підсистема, яка зберігає параметри маршрутизованих подій [3],  $prv (*)badd\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки додавання рядка у змісті візуальної таблиці,  $prv (*)btnAddDscClk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки додавання опису шаблону та штампу часу,  $prv (*)btnRememberBase\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки запам'ятовування назви бази даних у випадуючому списку назв,  $prv (*)btnShowTab\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки показати джерело формули. Залежно від контексту активної вкладки активує фокус вводу на відповідній вкладці або вікні редактора;  $prv (*)btnDToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки генерування формули опису доменів,  $prv (*)btnRToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки генерування формули опису відношення,  $prv (*)btntToMSSQL\_Clk (sdr \in obj, e \in RtdEvArgs)$  — обробник події кліку кнопки генерування формули опису кортежів,  $btnDToMSSQL\_Clk$ ,  $btnRToMSSQL\_Clk$ ,  $btntToMSSQL\_Clk$  зумовлюють функційний унітерм додавання формули до бібліотеки шаблонів,  $(*)btnGnPrvMsWhl (sdr \in obj, e \in MsWhlEvArgs)$  — обробник події прокрутки колеса миші при відкритій спливаючій підказці кнопки створення формули, де,  $e \in MsWhlEvArgs (MouseEventArgs)$  — типова відома підсистема, яка зберігає параметри події прокрутки колеса миші [3],  $prv (*)btnTblTXML\_TlTpOpn (sdr \in obj, e \in TlTpEvArgs)$  — обробник події відкриття спливаючої підказки кнопки створення формули відношення, де  $e \in TlTpEvArgs (ToolTipEventArgs)$  — типова відома підсистема, яка зберігає параметри події відкриття спливаючої підказки [3],  $prv (*)btnDmnTXML\_TlTpOpn (sdr \in obj, e \in TlTpEvArgs)$  — обробник події відкриття спливаючої підказки кнопки створення формули доменів,  $prv (*)btnTplTXML\_TlTpOpn (sdr \in obj, e \in TlTpEvArgs)$  — обробник події відкриття спливаючої підказки кнопки створення формули кортежів.

Текстові поля є засобами інтерфейсу, за допомогою яких у систему можна передавати дані з клавіатури. Опрацювання подій зміни змісту, отримання полем фокусу вводу, прокрутки колеса миші натискання клавіш на клавіатурі є тими функційними унітермами, у яких передані дані опрацьовані та використані в алгоритмах. Функціональний унітерм опрацювання подій текстових полів ( $H_2$ ) має вигляд формули (35). У ній  $prv (*)tbxTableProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs)$  — обробник події зміни змісту текстових полів, які встановлюють кількісні параметри відношення, де  $e \in TxtChngEvArgs (TextChangingEventArgs)$  — типова відома підсистема, яка зберігає параметри

події зміни змісту текстового поля [3],  $prv (*)tbxDomainProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs)$  — обробник події зміни змісту текстових полів які встановлюють кількісні параметри доменів,  $prv (*)tbxTupleProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs)$  — обробник події зміни змісту текстових полів, які встановлюють кількісні параметри кортежів,  $prv (*)bodyTablesTextbox\_MsWhl (sdr \in obj, e \in MsWhlEvArgs)$  — обробник події прокрутки колеса миші у текстових полях змісту візуальних таблиць,  $(*)ScrlTbxMsWhl (sdr \in obj, e \in MsWhlEvArgs)$  — обробник події прокрутки колеса миші у інших текстових полях вікна,  $prv (*)tbxNewTemplateName\_MsWhl (sdr \in obj, e \in MsWhlEvArgs)$  — обробник події прокрутки колеса миші у текстовому полі завдання назви шаблону,

$$H_2 = \left[ \begin{array}{l} prv (*)tbxTableProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs) \\ \vdots \\ prv (*)tbxDomainProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs) \\ \vdots \\ prv (*)tbxTupleProperty\_TxtChng (sdr \in obj, e \in TxtChngEvArgs) \\ \vdots \\ prv (*)bodyTablesTextbox\_MsWhl (sdr \in obj, e \in MsWhlEvArgs) \\ \vdots \\ (*ScrlTbxMsWhl (sdr \in obj, e \in MsWhlEvArgs) \\ \vdots \\ prv (*)tbxNewTemplateName\_MsWhl (sdr \in obj, e \in MsWhlEvArgs) \\ \vdots \\ prv (*)tbxAADescription\_MsWhl (sdr \in obj, e \in MsWhlEvArgs) \\ \vdots \\ prv (*)tbxNewTemplateName\_GtFcs (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ prv (*)tbxAADscrKUp (sdr \in obj, e \in KEvArgs) \\ \vdots \\ prv (*)tbxNwTmplNm\_MsDbIcIk (sdr \in obj, e \in MsBtnEvArgs) \\ \vdots \\ pbl (*)tbxNwTmplNmTScrSh() \end{array} \right], \quad (35)$$

де  $prv (*)tbxAADescription\_MsWhl (sdr \in obj, e \in MsWhlEvArgs)$  — обробник події прокрутки колеса миші у текстовому полі додаткового опису та штампа часу шаблону,  $prv (*)tbxNewTemplateName\_GotFcs (sdr \in obj, e \in RtdEvArgs)$  — обробник події прокрутки колеса миші у текстовому полі завдання назви шаблону,  $prv (*)tbxAADscrKUp (sdr \in obj, e \in KEvArgs)$  — обробник події натискання клавіш Enter клавіатури у полі додаткового опису шаблону, де,  $e \in KEvArgs$  (*KeyEventArgs*) типова відома підсистема, яка зберігає параметри події натискання клавіш на клавіатурі [3],  $prv (*)tbxNwTmplNm\_MsDbIcIk (sdr \in obj, e \in MsBtnEvArgs)$  — обробник події подвійного кліку у полі введення назви нової формули для додавання її до бібліотеки шаблонів, де  $e \in MsBtnEvArgs$

(*MouseButtonEventArgs*) — типова відома підсистема, яка зберігає параметри подій на миші [3].

Для подання користувачеві змісту бібліотеки шаблонів використовується список. Функціональні унітерми опрацювання подій списку шаблонів подано формулою  $L_1$ .

$$L_1 = \begin{bmatrix} prv (*)TmplLstUI\_MsDbIcIk (sdr \in obj, e \in MsBtnEvArgs) \\ \vdots \\ prv (*)GtTmplLst() \\ \vdots \\ prv (*)TmplLstUI\_SelCng (sdr \in obj, e \in SelCngEvArgs) \end{bmatrix}, \quad (36)$$

де  $prv (*)GtTmplLst()$  — функційний унітерм, який звертається до бази з назвою, що обрана у полі *cbxDaTaseName* та отримує список шаблонів, який відображається у елементі *TemplatesListUI* відомого типу *%ListView*,  $prv (*)TmplLstUI\_MsDbIcIk (sdr \in obj, e \in MsBtnEvArgs)$  — обробник події подвійного кліку у списку назв шаблонів,  $prv (*)TmplLstUI\_SelCng (sdr \in obj, e \in SelCngEvArgs)$  — обробник події зміни наявного елемента у списку назв шаблонів, де  $e \in SelCngEvArgs$  (*SelectionChangedEventArgs*) — типова відома підсистема, яка зберігає параметри події зміни наявного елемента у списку [3].

Функціональний унітерм опрацювання подій багатотекстових графічних елементів ( $C_1$ ) описано формулою (37). Багатотекстовий графічний елемент є текстовим полем з випадającym списком елементів, описується відомою графічною підсистемою *%ComboBox*. Його використання є доцільним, якщо дані мають обмежену кількість значень та задаються користувачем неодноразово, наприклад назви бібліотек, джерела даних для формул.

$$C_1 = \begin{bmatrix} pbl (*)StSrcIdx (idx \in int) \\ \vdots \\ prv (*)cbxDBNm\_SlcCng (sdr \in obj, e \in SelCngEvArgs) \end{bmatrix}, \quad (37)$$

де  $pbl (*)StSrcIdx (idx \in int)$  — функційний унітерм, який встановлює наявний багатотекстовий елемент *cbxTemplateID* за отриманим індексом *idx*, де, *pbl* — модифікатор доступу, який робить функційний унітерм публічним,  $idx \in int$  — змінна від 0 до 2, задає номер елемента — джерело формул, *int* (*Integer32*) — відомий системний тип, який надає у пам'яті цілі числа,  $prv (*)cbxDBNm\_SlcCng (sdr \in obj, e \in SelCngEvArgs)$  — обробник події зміни наявного елемента у багатотекстовому елементі, який зберігає назви бібліотек шаблонів.

Генерування формул опису на основі введених користувачем параметрів здійснюється окремою групою функційних унітермів. Функціональні унітерми генерування об'єктів опису формул алгебри алгоритмів описується формулою (38).

$$A_1 = \left[ \begin{array}{l} \text{prv} (*) \text{btnTblTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs}) \\ \vdots \\ \text{prv} (*) \text{btnDmnTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs}) \\ \vdots \\ \text{prv} (*) \text{btnTplTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs}) \\ \vdots \\ \text{prv} (Sq) \text{AdUntTHSq} (s1 \in Sq, u \in \underline{Unt}) \\ \vdots \\ \text{prv} (Sq) \text{AdUntTVSq} (s1 \in Sq, u \in \underline{Unt}) \\ \vdots \\ \text{prv} (Sq) \text{AdAgTrmTVSq} (s1 \in Sq, ag \in \underline{AgTm}) \\ \vdots \\ \text{prv} (Sq) \text{AdAgtTHSq} (s1 \in Sq, ag \in \underline{AgTm}) \end{array} \right), \quad (38)$$

де  $\text{prv} (*) \text{btnTblTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs})$  — функційний унітерм, який за даними, що наявні у візуальній таблиці створює формулу опису відношення, є також обробником події кліку кнопки «Створити відношення»,  $\text{prv} (*) \text{btnDmnTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs})$  — функційний унітерм, який за даними, що є у візуальній таблиці створює формулу опису доменів бази даних, є також обробником події кліку кнопки «Створити домени у базі даних»,  $\text{prv} (*) \text{btnTplTXML\_Clk} (sdr \in obj, e \in \underline{RtdEvArgs})$  — функційний унітерм який за даними, що є у візуальній таблиці створює формулу опису змісту відношення (кортежів), є також обробником події кліку кнопки «Створити кортежі»,  $\text{prv} (Sq) \text{AdUntTHSq} (s1 \in Sq, u \in \underline{Unt})$  — функційний унітерм повертає нове горизонтальне секвентування, яке впорядковує операцію секвентування  $s1$  та унітерм  $u$ , де  $s1 \in Sq$  — об'єкт операції секвентування, який є першим унітером нового секвентування,  $Sq$  (*Sequence*) — підсистема редактора @R, що описує операцію секвентування та базові дії над нею у формулах,  $u \in \underline{Unt}$  — об'єкт унітерму, який є другим унітером нового секвентування,  $\underline{Unt}$  (*Uniterm*) — підсистема, що описує у комп'ютерній системі унітерми формул алгебри алгоритмів,  $\text{prv} (Sq) \text{AdAgTrmTVSq} (s1 \in Sq, ag \in \underline{AgTm})$  — функційний унітерм повертає нове вертикальне секвентування, яке впорядковує операцію секвентування  $s1$  та операцію агрегування  $ag$ , де  $s1 \in Sq$  — об'єкт операції секвентування, який є першим унітером нового секвентування,  $ag \in \underline{AgTm}$  — об'єкт операції агрегування, що є другим унітером нового секвентування,  $\underline{AgTm}$  (*AggregateTerm*) — підсистема редактора @R, яка описує операцію агрегування та базові дії з нею у формулах,  $\text{prv} (Sq) \text{AdAgtTHSq} (s1 \in Sq, ag \in \underline{AgTm})$  — функційний унітерм повертає нове горизонтальне секвентування, яке впорядковує операцію секвентування  $s1$  та операцію агрегування  $ag$ , де  $s1 \in Sq$  — об'єкт операції секвентування, який є першим унітером нового секвентування,  $ag \in \underline{AgTm}$  — об'єкт операції агрегування, який є другим унітером нового секвентування,  $\text{prv} (Sq) \text{AdUntTVSq} (s1 \in Sq, u \in \underline{Unt})$  —

функційний унітерм, що повертає нове вертикальне секвентування, яке впорядковує операцію секвентування  $sl$  та унітерм  $u$ , де  $sl \in Sq$  — об'єкт операції секвентування, який є першим унітером нового секвентування,  $u \in Unt$  — об'єкт унітерму, який є другим унітером нового секвентування.

Бібліотека шаблонів є сховищем побудованим на основі реляційної СУБД та реалізуючим чотири базових функції зчитування, додавання, вилучення, оновлення даних з формулами. Функціональні ( $D_1$ ) унітерми взаємодії з бібліотекою шаблонів описуються формулою (39).

$$D_1 = \left[ \begin{array}{l} \left[ \begin{array}{l} pbl (*)UsTmpClk (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ prv (*)UpdTmpClk (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ pbl (*)AdTmpClk (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ prv (*)DltTmpClk (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ pbl sttc (\%AACnv)RndTmpCnv (AlgNm \in str) \\ \vdots \\ pbl sttc (\%Img)GtLnkFmlAsVctImg (Lnk \in str) \\ \vdots \\ pbl sttc (\%Img)GtLnkFmlAsRstImg (Lnk \in str) \\ \vdots \end{array} \right] \\ \vdots \\ \left[ \begin{array}{l} pbl sttc (bool)UsLnkTpl (Lnk \in str) \\ \vdots \\ pbl sttc (*)AdLnkFml (Lnk \in str) \\ \vdots \\ pbl sttc (*)UpdLnkFml (Lnk \in str) \\ \vdots \\ pbl sttc (byte [])GtDBCpFml (Lnk \in str) \end{array} \right] \end{array} \right] \quad (39)$$

де  $pbl (*)UsTmpClk (sdr \in obj, e \in RtdEvArgs)$  — функційний унітерм, який отримує з бібліотеки шаблонів формулу, назву якої вибрано у списку шаблонів  $TemplatesListUI$ , та використовує її у наявній формулі редактора  $@R$ ,  $prv (*)UpdTmpClk (sdr \in obj, e \in RtdEvArgs)$  — функційний унітерм, який для наявної у бібліотеці шаблонів формули, що вибрана у списку шаблонів  $TemplatesListUI$ , оновлює додатковий опис,  $pbl (*)AdTmpClk (sdr \in obj, e \in RtdEvArgs)$  — функційний унітерм, який додає до бібліотеки шаблонів вибрану у редакторі  $@R$  формулу з назвою, що задано у текстовому полі  $tbxNewTemplateName$ ,  $prv (*)DltTmpClk (sdr \in obj, e \in RtdEvArgs)$  — функційний унітерм видаляє формулу, яка обрана у списку шаблонів  $TemplatesListUI$ , якщо вона існує у наявній бібліотеці шаблонів,  $pbl sttc (\%Img)GtLnkFmlAsVctImg (Lnk \in str)$  — функційний унітерм, який отримує з бібліотеки шаблонів формулу, виконує її візуалізацію

як векторного зображення та повертає як графічний об'єкт *Image*, де *sttc* — модифікатор статичного члена підсистеми, який надає функційному унітерму належність до *@BDMaster*, а не до його екземплярів, *%Img* — відома підсистема, яка представляє зображення у документах та інтерфейсі користувача [3], *Lnk* ∈ *str* — текстова назва формули, яку треба візуалізувати як векторне зображення, *pbl sttc (%Img)GtLnkFmlAsRstImg (Lnk* ∈ *str)* — функційний унітерм, який отримує з бібліотеки шаблонів формулу, виконує її візуалізацію як растрового зображення та повертає як графічний об'єкт *Image*, де *Lnk* ∈ *str* — текстова назва формули, яку треба візуалізувати як растрове зображення.

*pbl sttc (bool)UsLnkTpl (Lnk* ∈ *str)* — функційний унітерм, який із бібліотеки шаблонів передає до редактора формулу з назвою *Lnk*, *pbl sttc (\*)AdLnkFml (Lnk* ∈ *str)* — функційний унітерм, що зберігає до бібліотеки шаблонів наявну формулу редактора під назвою *Lnk*, *pbl sttc (\*)UpdLnkFml (Lnk* ∈ *str)* — функційний унітерм, який у бібліотеці шаблонів оновлює формулу з назвою *Lnk* на ту, що є у редакторі наявною, *pbl sttc (byte[])GtDBCpFml (Lnk* ∈ *str)* — функційний унітерм, який із бібліотеки шаблонів повертає формулу з назвою *Lnk* як масив байтів, де *byte[]* відомий тип, який є динамічним масивом байтів, *pbl sttc (%AACnv)RndTplCnv (AlgNm* ∈ *str)* — функційний унітерм, який виконує візуалізацію формули на графічному об'єкті *AACnv*, де *AACnv* робоча область редактора *@R* наслідує від відомого типу *%Canvas*, який є візуальним елементом, що розміщує інші графічні елементи, позиціонуючи їх декартовими координатами (*x*, *y*), *AlgNm* ∈ *str* — назва формули.

Статичні функційні унітерми *GtLnkFmlAsVctImg*, *GtLnkFmlAsRstImg*, *UsLnkTpl*, *AdLnkFml*, *UpdLnkFml*, *GtDBCpFml*, *RndTplCnv* є публічними, та використовуються іншими підсистемами для роботи зі сховищем шаблонів. Вони мають загальний текстовий параметр, який є назвою формули.

Візуальна таблиця є базовим засобом інтерфейсу користувача комп'ютерної системи синтезу формул, описується відомою графічною підсистемою *%Table*. Параметри у її комірках є входними для алгоритмів генерації формул алгебри алгоритмів.

Функціональний унітерм генерування візуальних таблиць ( $T_1$ ) описується формулою (40).



$$T_1 = \left[ \begin{array}{l} pbl(*)AdDmnPrp() \\ \vdots \\ pbl(*)AdAtbPrp() \\ \vdots \\ pbl(*)AdTplPrp() \\ \vdots \\ pbl(*)RwTblInc (thisTbl \in \%Tbl) \\ \vdots \\ pbl(*)CrTbls (thisTbl \in \%Tbl) \\ \vdots \\ pbl(*)CrDmn (thisTbl \in \%Tbl) \\ \vdots \\ pbl(*)CrTpls (thisTbl \in \%Tbl) \\ \vdots \\ prv(*)RstrTblVls (TblHst \in \underline{str}[,], tbl \in \%Tbl) \\ \vdots \\ prv(*)RstrDmnVls (tbl \in \%Tbl) \\ \vdots \\ prv(*)RstrTplsVls (tbl \in \%Tbl) \\ \vdots \\ prv(*)tbxDRtTabs_GtFcs (sdr \in obj, e \in \underline{RtdEvArgs}) \\ \vdots \\ prv(\underline{str})GtTxfBx (g \in \underline{int}, r \in \underline{int}, c \in \underline{int}, tbl \in \%Tbl) \end{array} \right. , \quad (40)$$

де  $pbl(*)AdDmnPrp()$  — функційний унітерм, який задає значення словника типів доменів *DomainTypesList* та запам'ятовує значення комірок візуальної таблиці опису доменів у масив,  $pbl(*)AdAtbPrp()$  — функційний унітерм, який задає назви властивостей атрибутів відношення у словнику *Attribute Properties List* та запам'ятовує значення комірок візуальної таблиці опису заголовку відношення у масив,  $pbl(*)AdTplPrp()$  — функційний унітерм, який задає значення словника назв кортежів *TupleAttributesList* та запам'ятовує значення комірок візуальної таблиці опису змісту відношення у масив,  $pbl(*)RwTblInc (thisTbl \in \%Tbl)$  — функційний унітерм, який виконує нумерацію рядків візуальної таблиці, яку отримує у параметрі  $thisTbl$ , де,  $thisTbl \in \%Tbl$  — візуальна таблиця є відомою графічною підсистемою, яка дозволяє подавати табличні дані в інтерфейсі користувача,  $pbl(*)CrTbls (thisTbl \in \%Tbl)$  — функційний унітерм, який виконує побудову візуальної таблиці  $thisTbl$  опису заголовку відношення,  $pbl(*)CrDmn (thisTbl \in \%Tbl)$  — функційний унітерм, який виконує побудову візуальної таблиці  $thisTbl$  опису доменів бази даних,  $pbl(*)CrTpls (thisTbl \in \%Tbl)$  — функційний унітерм, що виконує побудову візуальної таблиці  $thisTbl$  опису змісту відношення. Функційні унітерми *CrDmn*, *CrTbls*, *CrTpls* виконують конструювання всіх динамічних частин інтерфейсу користувача та засобів керування ними,  $prv(*)RstrTblVls (TblHst \in \underline{str}[,], tbl \in \%Tbl)$  — функційний унітерм, який виконує відновлення

значень комірок візуальної таблиці  $tbl$  опису заголовка відношення із масиву  $TblHst$ , що буде отримано як перший параметр,  $prv (*)RstrDmnVls (tbl \in \%Tbl)$  — функційний унітерм, який виконує відновлення значень комірок візуальної таблиці  $tbl$  опису доменів з текстового масиву,  $prv (*)RstrTplsVls (tbl \in \%Tbl)$  — функційний унітерм, який виконує відновлення значень комірок візуальної таблиці  $tbl$  опису змісту відношення із текстового масиву,  $prv (str)GtTxfBx (g \in int, r \in int, c \in int, tbl \in \%Tbl)$  — функційний унітерм, який повертає текстове значення комірки візуальної таблиці  $tbl$ , положення якої задається числовими параметрами  $g, r, c$ , де  $g \in int$  — номер групи рядків,  $r \in int$  — номер рядка,  $c \in int$  — номер стовпчика,  $tbl \in \%Tbl$  візуальна таблиця, у якій шукається комірка.

Вікно є базовим елементом інтерфейсу користувача у сучасних операційних системах та його існування є основою життєвого циклу всієї підсистеми. Вкладки є основними компоновальними елементами графічного вікна, їх активність означає режими роботи комп'ютерної системи синтезу формул. Універсальні події — події, що виникають у різних об'єктах одного типу, але опрацьовуються одним загальним обробником функціональної підсистеми вікна. Глобальні події — події, які означають функціонування всіх елементів вікна комп'ютерної системи, наприклад, зміна розміру, закриття. Функціональні унітерми опрацювання універсальних та глобальних для вікна та вкладок ( $W_1$ ) подій, що описується формулою (41).

$$W_1 = \left[ \begin{array}{l} prv (*)MstClsd (sdr \in obj, e \in EvArgs) \\ \vdots \\ prv (*)TbDmnClk (sdr \in obj, e \in MsBtnEvArgs) \\ \vdots \\ prv (*)TbTplsClk (sdr \in obj, e \in MsBtnEvArgs) \\ \vdots \\ prv (*)TbTbIsClk (sdr \in obj, e \in MsBtnEvArgs) \\ \vdots \\ prv (*)tbxDRtTabs_GotFocus (sdr \in obj, e \in RtdEvArgs) \\ \vdots \\ prv (*)MstSzChng (sdr \in obj, e \in SzCngEvArgs) \\ \vdots \\ prv (*)TlTpHlp_TlTpOpnn (sdr \in obj, e \in TlTpEvArgs) \\ \vdots \\ prv (*)MstKDown (sdr \in obj, e \in KEvArgs) \end{array} \right), \quad (41)$$

де  $prv (*)MstClsd (sdr \in obj, e \in EvArgs)$  — обробник події закривання вікна, виконує зберігання параметрів сесії користувача у масиви та конфігураційні файли,  $prv (*)TbDmnClk (sdr \in obj, e \in MsBtnEvArgs)$  — обробник події вибору вкладки «Домени»,  $prv (*)TbTplsClk (sdr \in obj, e \in MsBtnEvArgs)$  — обробник події вибору вкладки «Кортежі»,  $prv (*)TbTbIsClk (sdr \in obj, e \in MsBtnEvArgs)$  — обробник події вибору вкладки «Відношення». Функційні унітерми  $TbDmnClk$ ,

*TbTplsClk*, *TbTbIsClk* налаштовують ширину робочого простору інтерфейсу користувача, *prv (\*)tbxDRTabs\_GotFocus (sdr∈obj, e∈RtdEvArgs)* — обробник події отримання фокусу вводу даних вкладками. Виконує приєднання обробника події прокрутки колеса миші *ScrolledTextbox\_MouseWheel*, *prv (\*)MstSzChng (sdr∈obj, e∈SzCngEvArgs)* — опрацьовує зміни вікна та робить перекомпонування контейнерів з елементами управління інтерфейсу користувача, *prv (\*)TlTpHlp\_TlTpOpnn (sdr∈obj, e∈TlTpEvArgs)* — обробник події відкриття спливаючої підказки, яка містить довідку за засобами інтерфейсу користувача, *prv (\*)MstKDown (sdr∈obj, e∈KEvArgs)* — обробник подій натискання кнопок клавіатури у вікні майстра. Опрацьовує натискання гарячих клавіш та виконує відповідні до введених користувачем комбінацій дії з формулами.

Робота системи синтезу формул починається із запуску конструктора  $K_1$  після закінчення роботи якого, система надає графічне вікно та очікує подій від його елементів, які можуть бути ініційовані користувачем. Конструктор описано формулою (42).

$$\begin{aligned}
 K_1 &= pbl\ BDMst() \\
 &= \\
 &\left( \begin{array}{l} IntCmpn() ; (AdDmnPrp() \\ \vdots \\ AdAtbPrp() ; (AdTplPrp() \\ \vdots \end{array} \right. \\
 &\left( \begin{array}{l} tbxDomainsNumber.Txt = MnFrm.MasterConfig[0].TStr() \\ \vdots \\ tbxDomainPropsNumber.Txt = MnFrm.MasterConfig[1].TStr() \\ \vdots \\ tbxTableAttributesNumber.Txt = MnFrm.MasterConfig[2].TStr() \\ \vdots \\ tbxAPropertyNumber.Txt = MnFrm.MasterConfig[3].TStr() \\ \vdots \end{array} \right. , \quad (42) \\
 &\left( \begin{array}{l} tbxTuplesNumber.Txt = MnFrm.MasterConfig[4].TStr() \\ \vdots \\ tbxTupleAttributesNumber.Txt = MnFrm.MasterConfig[5].TStr() \\ \vdots \end{array} \right. \\
 &\left( CrTbIs(tblR) ; CrDmn(tblD) \right. \\
 &\left( CrTpls(tblT) ; DmnRdr.Doc.PgWdt = 920 \right. \\
 &\left( tbxNwImplNm.Txt = DbCfgFrm.defaultNewAlgName ; fstst++ \\
 &\vdots \\
 &GtImplClk($, $)
 \end{aligned}$$

де *IntCmpn()* — відомий функційний унітерм, який ініціалізує графічне вікно системи у операційній системі; *tblR*, *tblD*, *tblT* — візуальні таблиці, які задають параметри для створення формул опису відношень, доменів, кортежів; *tbxDomainsNumber*, *tbxDomainPropsNumber*, *tbxTableAttributesNumber*,

*tbxAPropertyNumber*, *tbxTuplesNumber*, *tbxTupleAttributesNumber* — графічні елементи — текстові поля, керують кількістю рядків та стовпчиків візуальних таблиць *tblR*, *tblD*, *tblT*; *DmnRdr* — назва об'єкта відомого типу *FlwDocRdr*, у якому розміщено контейнер-документ з візуальними таблицями, де *Doc* — об'єкт документа, на якому розміщено візуальну таблицю; *PgWdt* — властивість ширини сторінки документа у пікселях; *DbCfgFrm* — підсистема редактора *@R*, у якій зберігається конфігурація бази даних, де *defaultNewAlgName* назва бази даних на замовчування, яка має використовуватися для бібліотеки шаблонів; *tbxNwTplNm* — назва нової формули; *TStr()* — відомий метод, який повертає об'єкт чи змінну як текст; *MnFrm* — головне вікно підсистеми редактора *@R*, де *MasterConfig* — публічна змінна — масив, яка зберігає останні задані параметри візуальних таблиць, коли вікна системи генерування *%BDMaster* не існує; *Txt* — відома властивість текстового поля *%TextBox* [3]. Вищезазначені графічні елементи описано у графічній частині моделі *%G*.

Генерацію формули опису доменів на основі параметрів заданих користувачем у візуальній таблиці *tblD* реалізує функційний унітерм *btnDmnTXML\_Clk()*. Він з'являється після натискання кнопки «Створити описи доменів для бази даних» на вкладці «Домени». Функційні унітерми *btnDmnTXML\_Clk()* і *Fu2*, описуються формулами (43) і (44), відповідно.

У формулі (43) *ag*  $\in$  *AgTm* — об'єкт операції агрегування; *e*  $\in$  *RtdEvArgs* — другий аргумент функційного унітерму, який описує параметри події; *i*, *j* — змінні циклічних операцій; *c* — кількість стовпчиків візуальної таблиці; *cj*, *ci* — ознаки повернення у цикл; *int* — Integer — системний тип цілих чисел; *left*  $\in$  *str* — змінна, яка зберігає ліву частину значення наявної властивості домену; *mf*  $\in$  *MainForm* — об'єкт класу *MainForm* — головного вікна системи; *obj* — відомий базовий тип *object*, від якого наслідуються усі інші типи; *prv* — модифікатор приватного доступу до функційного унітерму; *r*  $\in$  *int* — змінна *r*, яка містить кількість рядків у таблиці візуальної моделі, є параметром циклу; *right*  $\in$  *str* — змінна *right*, яка містить праву частину унітерму, подає наявні властивості домену; *s1*, *sv*  $\in$  *Sq* — змінні типу підсистеми операції секвентування до якої додаються унітерми властивості домену; *sdr*  $\in$  *obj* — змінна для збереження об'єкта у якому сталася подія; *str* (*String*) — текстовий тип даних; *thtbl* — скорочення назви (*thisable*) візуальної таблиці; *u*  $\in$  *Unt* — унітерм значення властивості домену; *AdAgTrmTVSq* — функційний унітерм (*AddAggregateTermToVSequence*), який приймає посилання на операції агрегування та секвентування, виконує вставлення агрегування на другу позицію секвентування та повертає модифіковане секвентування; *AdUntTHSq* — функційний унітерм (*AddUnitermToHSequence*), який приймає посилання на операції секвентування та унітерму, виконує вставлення унітерму на другу позицію секвентування з горизонтальною орієнтацією та повертає модифіковане секвентування; *AgTm* — підсистема (*AggregateTerm*), яка представляє операцію агрегування; *btnDmnTXML\_Clk* — функційний унітерм (*btnDomainToXML\_Click*), що виконує транслювання візуальної таблиці на формулу алгебри

алгоритмів та повертає формулу як об'єкт типу *Term*; *canvasFlag* — змінна-лічильник, яка враховує кількість викликів функціонального унітерму *btnDmnTXML\_Clk*; *Cls* — колекція (*Cells*) комірок таблиці; *Cnt* — властивість (*Count*) колекцій, повертає кількість елементів колекції; *GetTextfrBx* — функційний унітерм (*GetTextfromBox*), який повертає рядок з текстом, що є властивістю стандартного елемента керування *TextBox* та розташований у візуальній таблиці за отриманим у параметрах номера групи рядків, номера стовпчика та рядка; *MstTm* — об'єкт (*MasterTerm*), що зберігає згенеровану формулу і є статичною змінною підсистеми; *Orn* — перерахування (*Orientation*), яке кодує орієнтацію операції алгебри алгоритмів; *PrS()* — функційний унітерм (*PairSequence*) головного вікна редактора, виконує перегрупування секвенції парами; *PstfMst* — функційний унітерм (*PastefromMaster*) головного вікна редактора, що виконує розміщення згенерованої формули у кореневій формулі редактора, заміщуючи виділений унітерм; *RtdEvArgs* — системний об'єкт (*RoutedEventArgs*), який приймає параметри подій елементів інтерфейсу користувача; *RwGrp* — колекція (*RowGroups*) груп рядків у об'єктах типу *Table*; *Rws* — колекція рядків (*Rows*) у групі рядків таблиці; *Sq* — тип, який представляє операцію секвентування; *Tbl* — системний тип (*Table*), представляє елемент інтерфейсу користувача і використовується для візуалізації табличних даних; *tblD* — візуальна таблиця доменів (*tableD*), що означена у графічній частині моделі; *tA*, *tB* — властивості операції секвентування, які відповідно зберігають першого (*termA*) та другого (*termB*) унітермів операцій; *Unt* — підсистема опису унітермів (*Uniterm*); *Vrt* — значення вертикальної орієнтації (*Vertical*) операції.

$prv (*)btnDmnTXML\_Clk (sdr \in obj, e \in RtdEv.Args)$   
 $=$   
 $\left( \begin{array}{l} \overline{canvasFlag++}; *; (sdr \neq \$)-?; \left( \begin{array}{l} (thtbl \in \%Tbl) = tblD \\ \vdots \\ (left \in str) \end{array} \right) \\ \vdots \\ \uparrow; *; (mf = \$)-? \end{array} \right)$   
 $\left( \begin{array}{l} (right \in str) \quad ; \left( \begin{array}{l} (u \in Unt) \\ \vdots \\ (sv \in Sq) = new Sq(Sq.Orn.Vrt) \end{array} \right) \\ \vdots \\ (sl \in Sq) = new Sq() \end{array} \right)$   
 $\left( \begin{array}{l} (ag \in AgTm) = new AgTm() \\ \vdots \\ (c \in int) = thtbl.RwGrp[1].Rws[0].Cls.Cnt \\ \vdots \\ (r \in int) = thtbl.RwGrp[2].Rws.Cnt \\ \vdots \\ (j \in int) = 0 \end{array} \right)$   
 $\left( \begin{array}{l} \overline{MstTm = new Unt("Порожній вираз!"); *; (c=0 || r=0)-?} \\ \vdots \\ \overline{mf.PstfMst(MstTm); *; (canvasFlag > 0)-?} \\ \vdots \\ \uparrow \end{array} \right)$   
 $\left( \begin{array}{l} \overline{MstTm = new Unt("Некоректний опис домену!"); *; (c=1)-?} \\ \vdots \\ \overline{mf.PstfMst(MstTm); *; (canvasFlag > 0)-?} \\ \vdots \\ \uparrow \end{array} \right)$   
 $\left( \begin{array}{l} \vdots \\ (sl.tA = Unt(GtTxfBx(1, 0, 1, thtbl) + '=' + GtTxfBx(2, j, 1, thtbl) + '^')) \\ \vdots \\ (sl.tB = Unt(GtTxfBx(1, 0, 2, thtbl) + '=' + GtTxfBx(2, j, 2, thtbl) + '^')) \\ \vdots \end{array} \right)$   
 $Fu2$

(43)

```

Fu2=
┌
│   int i
│   ┌
│   │   int i = 3;
│   │   ┌
│   │   │   left = GtTxfBx(1, 0, i, thtbl) ; (u = Unt(left+'='+right+'')) *; (i < c)-?
│   │   │   right = GtTxfBx(2, j, i, thtbl)   (s1 = AdUntTHSq(s1, u)
│   │   │   └
│   │   │   i++; ci
│   │   └
│   │   ag.tA = new Unt('D['+GtTxfBx(2, j, 1, thtbl)+''])
│   └
│   ag.tB = PrS(s1); sv.tA = new Unt('DB['+ GtTxfBx(0, 0, 0, thtbl) + '']);
│
│   sv.tB = ag
│   ┌
│   │   int j
│   │   ┌
│   │   │   j = 1
│   │   │   ┌
│   │   │   │   s1 = new Sq()
│   │   │   │   s1.tA = Unt(GtTxfBx(1, 0, 1, thtbl) + '=' + GtTxfBx(2, j, 1, thtbl) + '\')
│   │   │   │   s1.tB = Unt(GtTxfBx(1, 0, 2, thtbl) + '=' + GtTxfBx(2, j, 2, thtbl) + '\')
│   │   │   │   ag = new AgTm(new Unt('D['+ GtTxfBx(2, j, 1, thtbl) + '']))
│   │   │   └
│   │   │   sv = AdAgTmTVSq(sv, ag)
│   │   │   ┌
│   │   │   │   int i
│   │   │   │   ┌
│   │   │   │   │   int i = 3
│   │   │   │   │   ┌
│   │   │   │   │   │   left = GtTxfBx(1, 0, i, thtbl) ; (u = Unt(left+'='+right+''))
│   │   │   │   │   │   right = GtTxfBx(2, j, i, thtbl)   (s1 = AdUntTHSq(s1, u)
│   │   │   │   │   │   └
│   │   │   │   │   │   i++; ci
│   │   │   │   │   └
│   │   │   │   │   *
│   │   │   │   │   (i < c)-?
│   │   │   │   └
│   │   │   │   ag.tB = PrS(s1)
│   │   │   └
│   │   │   j++; cj
│   │   └
│   │   *
│   │   (j < r)-?
│   └
│   MstTm = PrS(sv)
│   └
│   mf.PstfMst(MstTm); *; (canvasFlag > 0)-?
└
    
```

Отже, засобами алгебри алгоритмів формалізовано базовий елемент мови інформаційної технології — типізований шлях та такі його складові частини як адреса, ідентифікатор, вказівник типу об'єкта.

Модель типізованого шляху застосовується до опису ієрархії, у якій складові частини не є однаковими за типом та властивостями. Засобами алгебри алгоритмів формалізовано елемент мови інформаційної технології значення. Наведено застосування моделі типізованого шляху для опису об'єкта реляційної моделі та зроблено її верифікацію. Означення типізованого шляху дає основу для означення унтермів операцій над об'єктами баз даних.

Засобами алгебри алгоритмів побудовано математичну модель функційної частини інформаційної технології, яка описує створення і використання системи шаблонів у процесах синтезу доменів, заголовків відношень, значень відношень, збереження у бібліотеках шаблонів і синтезованих формул завдань. Описана у моделі комп'ютерна система підвищує рівень автоматизації процесів синтезу формул завдання на створення реляційних баз даних.

1. Дейт К. Дж. Введение в системы баз данных [пер. с англ.] / К. Дж. Дейт — М. : Вильямс, 2005. — 1328 с. 2. Кодд Э. Ф. Реляционная модель данных для больших совместно используемых банков данных [пер. с англ.] / Э. Ф. Кодд // Системы Управления Базами Данных # 1/1995. — 1970. — Вып. 13. 3. МакДональд М. WPF: Windows Presentation Foundation в .NET 3.0 для профессионалов / М. МакДональд. — М. : Вильямс, 2008. — 992 с. 4. Овсяк В. К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем / В. К. Овсяк // Доповіді Національної академії наук України. — 1996. — № 9. — С. 83–89. 5. Троэлсен Э. C# и платформа.NET. Библиотека программиста / Э. Троэлсен. — СПб. : Питер, 2005. — 796 с. 6. Owsiak W. Rozszerzenie algebry algorytmów / W. Owsiak, A. Owsiak // Pomiar, automatyka, kontrola. — 2010. — № 2. — S. 184–188.

## **МОДЕЛИ УНИТЕРМОВ И МОДЕЛЬ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ОПИСАНИЯ ЗАДАНИЯ НА СОЗДАНИЕ БАЗ ДАННЫХ**

*Рассмотрена модель типизированного пути как средства описания объектов реляционной модели. Построена математическая модель функциональной части информационной технологии, описывающая создание и использование системы шаблонов в процессах синтеза доменов, заголовков отношений, значений отношений, сохранения в библиотеках шаблонов и синтезированных формул и повышает уровень автоматизации процессов синтеза задания на создание реляционных баз данных.*

## **UNITERMS MODELS AND MODEL OF INFORMATION TECHNOLOGY DESCRIPTION OF TASK ON DATABASES CREATING**

*A model of the typed path as a means of relational model object describing . A mathematical model of the functional part of information technology, describes the creation and using of templates in the synthesis domains, relations header, relations tuples set, storing library of templates and synthetic formulas of creation task and increases the automation level of the synthesis of description of relational databases creating.*

*Стаття надійшла 30.09.2011*



УДК 655.59

*О. М. Величко, К. І. Савченко*  
*Видавничо-поліграфічний інститут*  
*НТУУ «КПІ»*

## **МІСЦЕ ГІБРИДНИХ ТЕХНОЛОГІЙ У ВПК УКРАЇНИ**

*Визначається ступінь розподілу друкарень по регіонах України та способах друку, які в них застосовуються. Систематизуються інсталяції друкарського обладнання, визначається місце гібридних технологій у ВПК України.*

### ***Гібридні технології, металізовані фарби, друкарня, технологічні процеси***

Гібридні технології в широкому розумінні цього нового терміна, усталеного в останнє десятиліття, тлумачиться як поєднання в одному виробничому процесі двох-трьох варіантів діаметрально протилежних типів технологічних рішень для створення оригінального або складного за конструкцією виробу. Тож об'єднання в одному виробничо-технологічному циклі друкування і оздоблення (лакування, тиснення, висікання тощо), друкування засобами плоского офсетного та високого, глибокого, трафаретного або струминного друку, матеріалів з різними фізико-хімічними властивостями — слід визначати як гібридні технології, а технічні засоби їх вирішення — як гібридне обладнання і гібридні процеси, що передбачає, наприклад, використання гібридних фарб у друкарсько-оздоблювальному процесі. У вітчизняних джерелах майже немає інформації щодо експлуатації гібридних технологій, обладнання і матеріалів на практиці. Даних щодо унормування параметрів перенесення і закріплення гібридних фарб на відбитках плоского офсетного друку і забезпечення довговічності їх експлуатаційних характеристик надзвичайно мало. Слід визначити наскільки активно ці технології впроваджуються в Україні. Для цього варто проаналізувати способи друку й технологічне оснащення, які використовуються в Україні, а також світові тенденції виробництва друкованої продукції.

Щодо відображення та аналізу ключових тенденцій, проблем і перспектив розвитку поліграфічної діяльності в Україні дослідження проводяться лише у ВПІ НТУУ «КПІ» та Української академії друкарства.

Зокрема відомими є напрацювання діячів Української академії друкарства [3], які проводять моніторинг поліграфічного ринку України, аналізують та систематизують ринок друкованої продукції.

Науковцями ВПІ НТУУ «КПІ» також проводилися дослідження ринкових концепцій поліграфічного виробництва, висвітлювалися проблеми та перспективи поліграфічної галузі [5, 7].

На світовому рівні відомими є праці [12, 14] стосовно розвитку способів