

Статистична модель аналізу текстів / М.В. Дідковська, Д.В. Старосуд // Нові технології. – 2011. – №2 (32) – С. 62 – 69. 4. К вопросу о методах анализа документов в информационной деятельности / Н.В. Махортова, И.А. Омеляненко, С.Г. Шапошникова и др. // Вісн. Східноукраїн. нац. ун-ту ім. В. Даля. – 2008. – №8 (126), Ч.1. – С. 359–362. 5. Крыгин М.Ю. Текст на естественном языке как объект статистического анализа / М.Ю. Крыгин // Бионика интеллекта. – 2010. – №1(72). – С. 75 – 82. 6. Мисуно И.С. Векторные и распределенные представления, отражающие меру семантической связи слов / Мисуно И.С., Рачковский Д.А., Слипченко С.В. // Математичні машини і системи. – 2005. – №3 – С. 50–66. 7. Об одном методе статистической фильтрации текстовой информации: материалы Междунар. науч. конф. “Современные информационные технологии и письменное наследие: от древних рукописей к электронным текстам”. – Ижевск, 2006. – 126 с. 8. Павлишенко Б. Семантична кластеризація текстових документів методом k-середніх / Б. Павлишенко // Вісн. Нац. ун-ту «Львівська політехніка». – 2011. – № 710. – С. 215–218. 9. Павлишенко Б. Групування текстових даних на основі моделі семантичного контексту / Б. Павлишенко // Восточно-Европ. журн. передовых технологий. – 2011. – №5/2 (53). – С. 39 – 42. 10. Текстовая кластеризация алгоритмом ROCK : материалы XVII всероссийской науч.-методич. конф. [«Телематика’2010»]. [Электронный ресурс] / Савин. И.И. – 2010 – Режим доступа: <http://tm.ifmo.ru/tm2010/src/263e.pdf>. 11. Christopher D. Manning. Introduction to information retrieval / Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze // Cambridge University Press. – Cambridge, 2008.

ПРЕДСТАВЛЕНИЕ СЕМАНТИКИ ТЕКСТА ПОСРЕДСТВОМ СТАТИСТИЧЕСКИХ ПОКАЗАТЕЛЕЙ

Проанализированы методы представления текстовых документов в информационно-поисковых системах. Описаны возможности представления содержания текста посредством статистических показателей.

PRESENTATION OF TEXT SEMANTIC THROUGH ITS STATISTICAL INDICATORS

It was analyzed methods represent text documents in information retrieval systems and the possibility of meaning representation of text through its statistics.

Стаття надійшла 14.11.2012

УДК 004

М. Козелко

Українська академія друкарства

МОДЕЛЬ ФУНКЦІОНУВАННЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕДАКТОРА ГРАФІЧНИХ УНІТЕРМІВ

Засобами алгебри алгоритмів описано модель функціонування інструментальних засобів редактора графічних унітермів.

Унітерм, абстракція, графічний інтерфейс, модель, система, проект

Абстрактні унітерми дають можливість відмовитися від строго типізованих елементів на етапі розроблення інтерфейсу. Такий підхід універсальніший і гнучкіший, передбачає повторне використання спроектованих інтерфейсів у проєктах різних типів – таких, як Windows-програма [6] чи інтернет-сторінка й інших. Однак при компіляції готового проєкту всі абстрактні унітерми повинні бути конвертовані до типових графічних. У зв'язку з тим існує проблема коректного конвертування абстрактних унітермів до типових графічних із заданням усіх властивостей та методів їх опрацювання, а також дочірніх елементів абстрактного унітерму до типового графічного.

Модель функціонування інструментальних засобів графічних унітермів

Модель функціонування редактора графічних унітермів – це модель, що описує дії та їх послідовність, які користувач повинен виконати для побудови графічного інтерфейсу, а також реакцію редактора на виконання цих дій. Математичну модель інструментальних засобів побудови графічних унітермів описано з використанням алгебри алгоритмів [1–5] і зображено формулою

$$Sf = \overbrace{Af, Bf, Cf, Df, Ef, Ff}^{\quad}$$

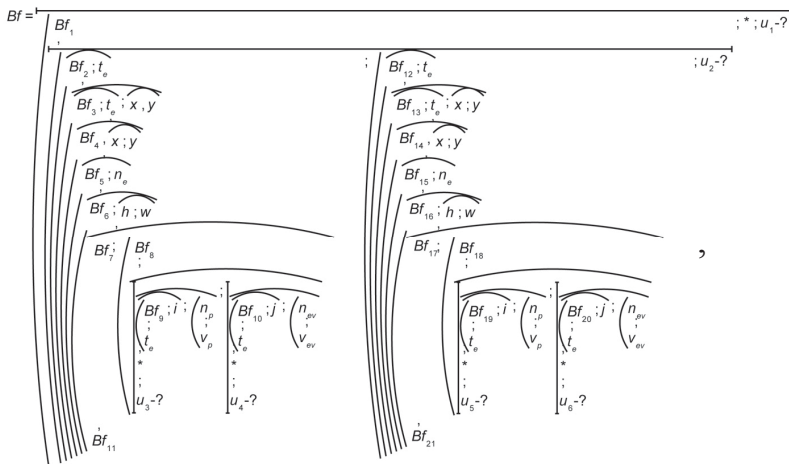
де Sf – модель інструментальних засобів побудови графічних унітермів; Af – підсистема створення унітерму-форми; Bf – підсистема генерування графічних та абстрактних елементів; Cf – підсистема опрацювання дій над графічними унітермами та надання їм функціональних особливостей; Df – підсистема конвертації абстрактних унітермів до типових графічних; Ef – підсистема генерування створеного проєкту до настільної чи веб-системи; Ff – підсистема компіляції створеного проєкту.

Підсистема Af призначена для створення нових унітермів-форм із заданням їх властивостей і подій. Підсистема Af описана такою формулою алгебри алгоритмів:

$$Af = \left[\begin{array}{l} Af_1 \quad ; * ; u_1 - ? \\ Af_2 \\ Af_3 \quad ; Af_4 ; u_2 - ? \\ n_f \\ h ; w \\ Af_5 \\ Af_6 ; Af_7 ; t_0 ; Af_8 ; Af_9 ; x ; y \\ Af_{10} \end{array} \right]$$

Для створення унітерму-форми потрібно в редакторі формул алгоритмів вибрати графічний унітерм (Af_1), після того згенерується унітерм-форма. Наступним кроком є задання назви нового унітерму-форми (Af_2). Якщо назву унітерму-форми задано, то після її перевірки (n_1) маємо генерування нового унітерму-форми (Af_3). Крім того, користувач отримує можливість припинити створення нової форми (умова u_2) і продовжити працювати з редактором формул алгебри алгоритмів (Af_4). Далі відбувається задання властивостей створеного унітерму-форми: висоти (h) і ширини (w). Для змоги створювати та розміщувати на унітермі-формі інших графічних унітермів створеному унітерму-формі задаються події генерування графічних унітермів (Af_5) і подія переміщення графічних унітермів (Af_6). Подія Af_6 виникає при переміщенні на створену унітерм-форму графічного позначення відповідного типу унітерму. При наведенні графічного позначення типу графічного елемента функціональним унітермом Af_7 описується тип f_e унітерму, який потрібно згенерувати. Функціональним унітермом Af_8 описується генерування обраного графічного унітерму. Розміщення згенерованого графічного унітерму описується функціональним унітермом Af_9 після закінчення переміщення графічного позначення на унітерм-форму. Згенерований графічний унітерм розміщується за координатами x та y . Подія Af_5 виникає при переміщенні графічного унітерму по унітермі-формі. Функціональним унітермом Af_{10} завершується опис генерування нового унітерму-форми і розміщення його в редакторі графічних унітермів для подальшої роботи над проектом.

Генерування графічних чи абстрактних унітермів описується підсистемою Bf , поданою у вигляді формули



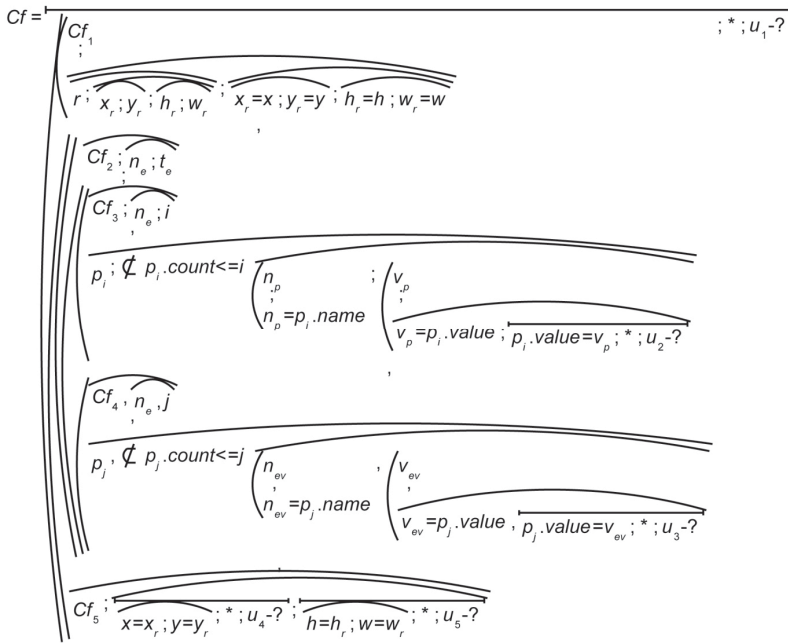
де u_1 – перевірка вибору користувачем графічного унітерму з набору можливих (Bf_1); u_2 – умова перевірки вибраного методу генерування графічних унітермів. Існує два методи створення користувачем графічного унітерму. У першому

випадку користувач може перемістити графічне позначення обраного унітерму з переліку елементів на унітерм-форму, що приведе до генерування графічного унітерму обраного типу та опису функціональним унітермом Bf_2 . Другий спосіб генерування вибраного графічного унітерму здійснюється після виконання подвійного цоку на графічному позначенні вибраного унітерму, що описується Bf_{12} ; t_e – визначення типу обраного користувачем унітерму для подальшого його генерування; x та y – координати розміщення графічного позначення відносно унітерму-форми; Bf_3 – функціональний унітерм ініціалізації створення нового екземпляра графічного унітерму з типом t_e ; Bf_4 – функціональний унітерм задання параметрів розміщення графічного унітерму на унітермі-формі за координатами x та y , попередньо визначеними функціональним унітермом Bf_3 ; Bf_5 – функціональний унітерм задання графічним унітермам власної назви (n_e), потрібної для ідентифікації графічних унітермів, і звертання до них у функціональній системі редактора графічних унітермів; Bf_6 – функціональний унітерм задання початкових розмірів графічного унітерму, його висоти h та ширини w ; Bf_7 – функціональний унітерм задання графічному унітерму події вибору користувачем графічного унітерму, необхідної для створення можливості редагувати властивості графічних унітермів після їх створення; Bf_8 – функціональний унітерм, який описує опрацювання події вибору графічного унітерму користувачем; Bf_9 – функціональний унітерм визначення типу вибраного графічного унітерму t_e і, відповідно, кількості доступних властивостей (i) графічних унітермів з типом t_e ; u_3 – умова перевірки кількості властивостей, потрібна для формування їх переліку, який складається з назви властивості n_p і відповідного значення властивості v_p ; Bf_{10} – функціональний унітерм визначення типу обраного графічного унітерму t_e , кількості подій j , притаманних типу t_e обраного графічного унітерму. Залежно від значення j формується перелік усіх доступних подій обраного графічного унітерму, котрий містить назву події n_{ev} і назву опрацювання методу цієї події v_{ev} ; Bf_{11} – функціональний унітерм закінчення генерування графічного унітерму й розміщення його на унітермі-формі.

Генерування графічного унітерму методом подвійного цоку по графічному позначенні унітерму відбувається аналогічно методу перетягування графічного позначення на унітерм-форму, за винятком того, що графічний унітерм розміщується на унітермі-формі за вільними від інших графічних унітермів координатами, а не за координатами розташування графічного позначення унітерму відносно унітерму-форми.

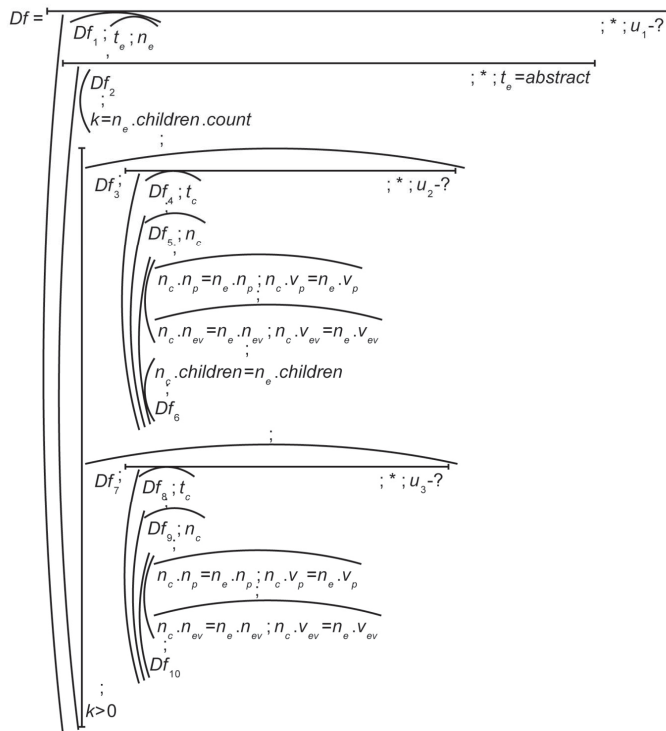
Редагування властивостей графічних унітермів та задання методів опрацювання їх подій реалізується підсистемою Cf . При виборі графічного унітерму на унітермі-формі функціональним унітермом Cf_j генерується унітерм виділення графічного унітерму r . Унітерм виділення r характеризується певними властивостями, зокрема: положенням на унітермі-формі – координатами x_r і y_r , розміром унітерму, вибору – висоти h_r і довжини w_r . Для коректного відображення унітерму вибору його властивостям задаються початкові параме-

три, аналогічні відповідним параметрам вибраного графічного унітерму. Так, координатам розміщення унітерму виділення задаються значення $x_r=x$, $y_r=y$, а властивостям розміру – $h_r=h$, $w_r=w$. Перевірка вибору графічного унітерму на унітермі-формі відбувається за умовою u_1 -? Зважаючи на те, що різним типам графічних унітермів притаманні неоднакові властивості і події, функціональним унітермом Cf_2 визначається тип обраного графічного унітерму t_e . Функціональний унітерм Cf_2 визначає також власну назву n_e обраного графічного унітерму. Це потрібно для того, щоб провести редагування властивостей графічного унітерму саме з обраним графічним унітермом, а не з будь-яким іншим графічним унітермом спільного типу t_e . Залежно від вибраного графічного унітерму n_e функціональним унітермом Cf_3 описується визначення кількості (i) всіх характерних для обраного графічного унітерму властивостей. Функціональним унітермом Cf_3 формується перелік властивостей вибраного графічного унітерму p_i . Такий перелік складається з назви властивості n_p та відповідного значення властивості v_p . Формування переліку властивостей відбувається циклічно поки не виконається умова рівності кількості елементів переліку (p_i) до кількості (i) властивостей обраного графічного унітерму. Перевірка рівності цих величин здійснюється умовою $p_i.count \leq i$, де $count$ властивість, яка описує обчислення кількості елементів переліку p_i . Елементом n_p і v_p переліку p_i задаються відповідні значення обраного графічного унітерму, а власне $n_p = p_i.name$ та $v_p = p_i.value$, де $name$ – назва властивості, а $value$ – значення цієї властивості. При зміні користувачем значення властивості v_p (виконання умови u_2 -?) відбувається автоматичне присвоєння нового значення властивості обраного графічного унітерму ($p_i.value = v_p$). Функціональний унітерм Cf_4 описує обчислення кількості (j) подій вибраного графічного унітерму n_e і формує перелік подій p_j вибраного графічного унітерму n_e . Перелік подій p_j складається з назви події n_{ev} та методу опрацювання події v_{ev} . Формування переліку p_j відбувається в циклі поки виконується умова $p_j.count \leq j$, тобто кількість елементів переліку дорівнюватиме кількості подій, властивих обраному графічному унітерму. Елементом n_{ev} і v_{ev} переліку p_j задаються відповідні значення вибраного графічного унітерму ($n_{ev} = p_j.name$, $v_{ev} = p_j.value$), де $name$ – назва події, а $value$ – метод опрацювання цієї події. При зміні користувачем методу опрацювання події v_{ev} (виконання умови u_3 -?) відбувається автоматичне присвоєння нового методу опрацювання події вибраного графічного унітерму $p_j.value = v_{ev}$. Функціональний унітерм Cf_5 містить зміну положення відносно унітерму-форми та розміру вибраного графічного унітерму за допомогою миші. При зміні користувачем положення унітерму виділення r (умова u_4 -?) координатам розміщення вибраного графічного унітерму задаються відповідні значення виділеного унітерму ($x=x_r$, $y=y_r$). Зміна користувачем розмірів виділеного унітерму приводить до зміни розмірів вибраного графічного унітерму: $h=h_r$, $w=w_r$. Формула, якою описані процеси редагування властивостей графічних унітермів і задання методів опрацювання їхніх подій, має такий вигляд:

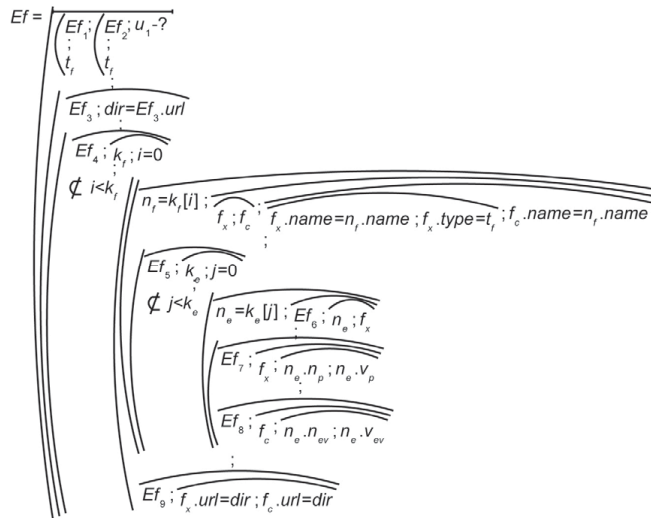


Процеси конвертації абстрактних унітермів до типових описуються підсистемою Df . Після вибору користувачем абстрактного унітерму на унітерміформі (виконання умови $u_1-?$) за функціональним унітермом Df_1 визначаються назва обраного унітерму n_e та його тип t_e . Оскільки операцію конвертування типів графічних унітермів можливо застосувати тільки до унітермів абстрактного типу, то здійснюється перевірка умови належності обраного унітерму до абстрактного типу ($t_e = abstract$). Якщо рівність $t_e = abstract$ виконується, функціональний унітерм Df_2 визначає кількість k дочірніх графічних унітермів обраного абстрактного унітерму ($k = n_e.children.count$), де $children$ властивість наявності дочірніх унітермів, $count$ – кількість дочірніх унітермів. Визначення кількості дочірніх графічних унітермів запобігає втраті цих унітермів при конвертації абстрактного унітерму. Залежно від кількості дочірніх графічних унітермів, розміщених на обраному абстрактному унітермі, формується перелік типів графічних унітермів, доступних для конвертації обраного абстрактного унітерму. Якщо $k > 0$, формується перелік Df_3 з графічних унітермів контейнерного типу. Після вибору користувачем потрібного типу графічного унітерму з переліку Df_3 (виконання умови $u_2-?$) за функціональним унітермом Df_4 визначається обраний тип графічного унітерму t_c . Відповідно до визначеного типу t_c за функціональним унітермом Df_5 генерується графічний унітерм n_c обраного типу. Для коректної конвертації абстрактного унітерму новому згенерованому унітерму n_c задаються всі властивості та їх значення обраного абстрактного унітерму ($n_c.n_p = n_e.n_p$, $n_c.v_p = n_e.v_p$), а також події та методи їх опрацювання (n_c .

$n_{ev} = n_e.n_{ev}$, $n_c.v_{ev} = n_e.v_{ev}$). Зважаючи на виконання умови $k > 0$, усі дочірні графічні унітерми обраного графічного унітерму n_e задаються дочірніми унітермами графічного унітерму n_c . Задання дочірніх унітермів графічного унітерму n_c відбувається за формулою $n_c.children = n_e.children$. Після успішного конвертування абстрактного унітерму до типового графічного щойно згенерований графічний унітерм за функціональним унітермом Df_6 розміщується на унітермі-формі. При виконанні умови $k = 0$ формується перелік з усіх доступних типів графічних унітермів Df_7 . Після вибору типу графічного унітерму з переліку Df_7 (виконання умови $u_3 - ?$) за функціональним унітермом Df_8 визначається обраний користувачем тип графічного унітерму t_c з переліку Df_7 . За функціональним унітермом Df_9 , відповідно до обраного типу t_c , генерується графічний унітерм n_c . Задаються властивості та їх значення $n_c.n_p = n_e.n_p$, $n_c.v_p = n_e.v_p$, а також події і методи їх опрацювання $n_c.v_{ev} = n_e.v_{ev}$. Розміщення конвертованого унітерму на унітермі-формі здійснюється за функціональним унітермом Df_{10} . Математична модель конвертації абстрактних унітермів до типових має вигляд такої формули алгебри алгоритмів:



Підсистема Ef реалізує генерування створеного проекту до настільної чи веб-системи. Математична модель даної підсистеми є наступною формулою алгебри алгоритмів:



У ній модель генерування та збереження створеного користувачем проекту описана вибором відповідного графічного унітерму в редакторі графічних унітермів. Залежно від вибраного типу для генерації створеного проекту (виконання умови $u_1-?$) за функціональним унітермом Ef_1 ініціалізується генерування настільного проекту або за функціональним унітермом Ef_2 – веб-проекту. Крім того, за функціональними унітермами Ef_1 і Ef_2 задається на генерування тип (t_f) обраного проекту. Функціональний унітерм Ef_3 призначений для відображення на унітермі-формі заданого шляху dir збереження проекту у файлової системі комп'ютера. Користувач вибирає унітерм-форму і задає шлях збереження $dir = Ef_3.url$. Зважаючи на те, що створений проект може бути багатівіконним функціональним унітермом, за функціональним унітермом Ef_4 встановлюється кількість (k_f) створених унітермі-форм. Залежно від значення k_f у циклі визначаються всі створені унітермі-форми ($n_f = k_f[i]$). Для кожного унітерму-форми n_f генерується два файли опису унітерму-форми f_x і f_c . Файл f_x слугує для опису створених унітермі-форм з усіма графічними унітермами, описує тип і назву унітерму, візуальні параметри, властивості та їх значення, задані події над унітермом. Файл f_c призначений для опису методів опрацювання подій графічних унітермів і функціональних особливостей унітермі-форми. Файлам опису задаються назви, ідентичні назві унітермі-форми ($f_x.name = n_f.name, f_c.name = n_f.name$), а файлу f_x – тип обраного проекту $f_x.type = t_f$. Щоб описати усі графічні унітерми, розміщені на унітермі-формі n_f , за функціональним унітермом Ef_5 визначається кількість графічних унітермів k_e , розташованих на унітермі-формі. Залежно від кількості графічних унітермів k_e формується цикл, де визначаються всі графічні унітерми $n_e = k_e[j]$, розміщені на унітермі-формі f_e . За функціональним унітермом Ef_6 здійснюється запис опису графічного унітерму n_e у

файл f_x . Запис властивостей $n_e.n_p$ та їхніх значень $n_e.n_v$ графічного унітерму n_e забезпечується за функціональним унітермом Ef_7 . Опис методів опрацювання подій графічних унітермів $n_e.v_{ev}$ виконується за функціональним унітермом Ef_8 . Запис проводиться у файл f_c . Згенеровані файли опису проекту зберігаються у файлової системі комп'ютера за функціональним унітермом Ef_8 , файли опису – у файлової системі за вказаним користувачем шляхом $f_x.url=dir, f_c.url=dir$.

Компіляція створеного проекту описується моделлю підсистеми Ff . Розпочинається після вибору користувачем відповідного графічного унітерму редактора графічних унітермів (умова u_1). При виконанні умови u_1 за функціональним унітермом Ff_1 ініціалізується компіляція проекту і відображається унітерм-форма (Ff_2), що вміщатиме проект (p), який потрібно компілювати. Визначається також місце розміщення проекту у файлової системі комп'ютера ($dir=Ff_2.url$). За функціональним унітермом Ff_3 здійснюється компіляція проекту p і генерується скомпільований проект (p_{ex}). Скомпільований проект записується у файлової системі за визначеним шляхом $p_{ex}.url=dir$ функціональним унітермом Ff_4 . За функціональним унітермом Ff_5 здійснюється запуск на виконання і відображення скомпільованого проекту p_{ex} . Даний процес описується такою формулою:

$$Ff = \left(\begin{array}{l} Ff_1 ; * ; u_1 - ? \\ Ff_2 ; dir = Ff_2.url ; p \\ Ff_3 ; \left(\begin{array}{l} p \\ p_{ex} \end{array} \right) \\ Ff_4 ; p_{ex}.url = dir \\ Ff_5 ; p_{ex} \end{array} \right)$$

З вищевикладеного впливають наступні висновки. Абстрактним унітермам притаманні дві ознаки: властивості та події унітерму. До перших відносяться всі властивості, характерні для графічних елементів: висота, ширина, назва, прозорість, активність тощо. Події представляють методи обробки дій користувача.

Абстрактні унітерми можна застосовувати на етапі розробки як основу майбутнього проекту невизначеного типу, перед компіляцією проекту слід вказати, чи це буде програма для операційної системи Windows, чи web-проект, чи окремий компонент для подальшого використання в інших проектах. Використання абстрактних унітермів дозволяє відмовитись від строго типізованих елементів при розробленні інтерфейсу. Такий підхід відзначається універсальністю, гнучкістю, передбачає повторне застосування спроектованих інтерфейсів у проектах різних типів.

1. Овсяк В.К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем / В.К. Овсяк // Доп. Нац. акад. наук України. – 1996. – № 9. – С.83–89. 2. Овсяк В.К. Алгоритми: аналіз методів, алгебра впорядкувань, моделі, моделювання. – Львів, 1996. – 132 с. 3. Owskiak A. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych / Owskiak W., Owskiak A., Owskiak J. // Opole: Politechnika Opolska, 2005. – 275 s. 4. Ovsyak V. The extended algebra of algorithms with multiconditional elimination / V. Ovsyak, A. Ovsyak // Вісн. Нац. ун-ту “Львівська політехніка”: Комп’ютерні науки та інформаційні технології. – 2010. – № 672. – С. 291 – 300. 5. Owskiak W. Rozszerzenie algebry algorytmów / W. Owskiak, A. Owskiak // Pomiar, automatyka, kontrola. – 2010. – № 2. – S.184–188. 6. Шилдт Герберт. С# 4.0: Полн. рук.; пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1056 с.

МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РЕДАКТОРА ГРАФИЧЕСКИХ УНИТЕРМОВ

Средствами алгебры алгоритмов описана модель функционирования инструментальных средств редактора графических унитермов.

MODEL OPERATION TOOLKITS EDITORS GRAPHIC UNITERMS

Means algebra algorithms described model of tools unitermiv graphic editor.

Стаття надійшла 04.10.2012

УДК 512.8

Р. В. Коляда

Українська академія друкарства

О. М. Мельник

Львівський національний університет ветеринарної медицини та біотехнологій імені С.З. Гжицького

ПРОСТОРОВІ МАТРИЦІ: АНАЛІЗ І ПЕРСПЕКТИВИ

Розглядаються різні математичні підходи до аналізу багатовимірних даних: класичний, векторно-матричний, тензорний, функціонального аналізу. Здійснено огляд стану та визначено перспективи розвитку матричного підходу до аналізу багатовимірних даних. Лінеаризовано матричний многочлен, коефіцієнтами якого є кубічні числові матриці.

Багатовимірні дані, просторова матриця, добуток матриць, транспонування матриці, лінеаризація

Розглядатимемо математичні дані, які в мовах програмування прийнято називати цілими чи дійсними константами або змінними. Будемо називати їх багатовимірними, якщо вони складаються з одного або більше чисел, з однієї або більше змінних. У даний час можна виділити декілька підходів до аналізу багатовимірних даних: класичний (скалярний); функціонального аналізу; векторно-матричний; тензорний; багатовимірно-матричний.