

**АЛГЕБРА АЛГОРИТМОВ С МНОГОЗНАЧНЫМ
ЭЛИМИНИРОВАНИЕМ**

Аксиоматическим методом дано определение алгебры алгоритмов с многозначным элиминированием.

ALGEBRA OF ALGORITHMS WITH MULTI ELIMINATION

Axiomatic method definitions given algebra of algorithms surgery multivalve elimination.

Стаття надійшла 09.10.2012

УДК 004.4'232

М. А. Козелко

Українська академія друкарства

**МОДЕЛЬ КОНВЕРТАЦІЇ АБСТРАКТНИХ ДО ТИПОВИХ
ГРАФІЧНИХ УНІТЕРМІВ АВТОМАТІВ**

Засобами алгебри алгоритмів описано створену модель побудови системи типових графічних унітермів, призначених для конвертації абстрактного унітерму.

Система, модель, графічний унітерм, абстрактний графічний унітерм, алгоритм

Інструментальні засоби інформаційних технологій і систем реалізуються графічними інтерфейсами. Можливі різні типи інструментальних засобів, наприклад, Windows або Web-проектів. Сам процес проектування інструментальних засобів творчий і складний. Особливо на початкових стадіях проектування є невідомими як загальний вигляд інструментальних засобів, так і його складові (графічні унітерми [3, 4, 5]) – кнопки, елементи альтернативного і безальтернативного вибору, текстові поля та інші. Для спрощення складності проектування інструментальних засобів доцільним є введення абстрактних графічних елементів – унітермів.

Використання абстрактних унітермів обумовлює відмову від строго типізованих елементів при розробленні інтерфейсу і передбачає повторне застосування спроектованих інтерфейсів для проектів різних типів, таких, як Windows програма [4, 5] чи інтернет-сторінка [1] й інших. Однак при компіляції готового проекту всі абстрактні унітерми повинні бути конвертовані до типових графічних. З огляду на це існує проблема коректного конвертування абстрактних унітермів до типових графічних із заданням усіх властивостей і методів їх опрацювання, а також дочірніх елементів абстрактного унітерму до типового графічного.

Схематична модель системи типових графічних унітермів

У спростованій схематичній моделі (див. рисунок) система типів L утворюється з s графічних унітермів контейнерного типу, тобто таких, які можуть містити інші графічні унітерми. Для кожного i -го графічного унітерму задається поле l_i , утворене назвою l_n і графічним позначенням l_i . Кожному полю задаються його висота $l.h$ і ширина $l.w$, а кожному полю l — подія вибору його користувачем для подальшого конвертування абстрактного унітерму до вибраного графічного типу.

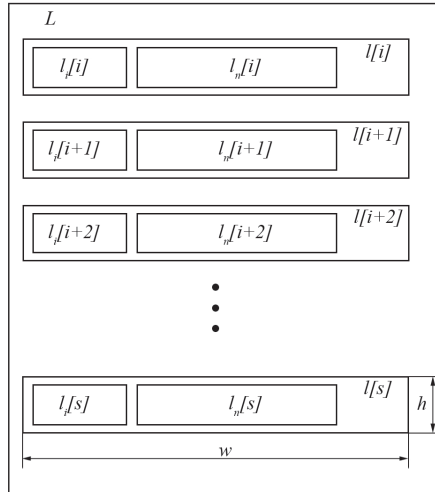
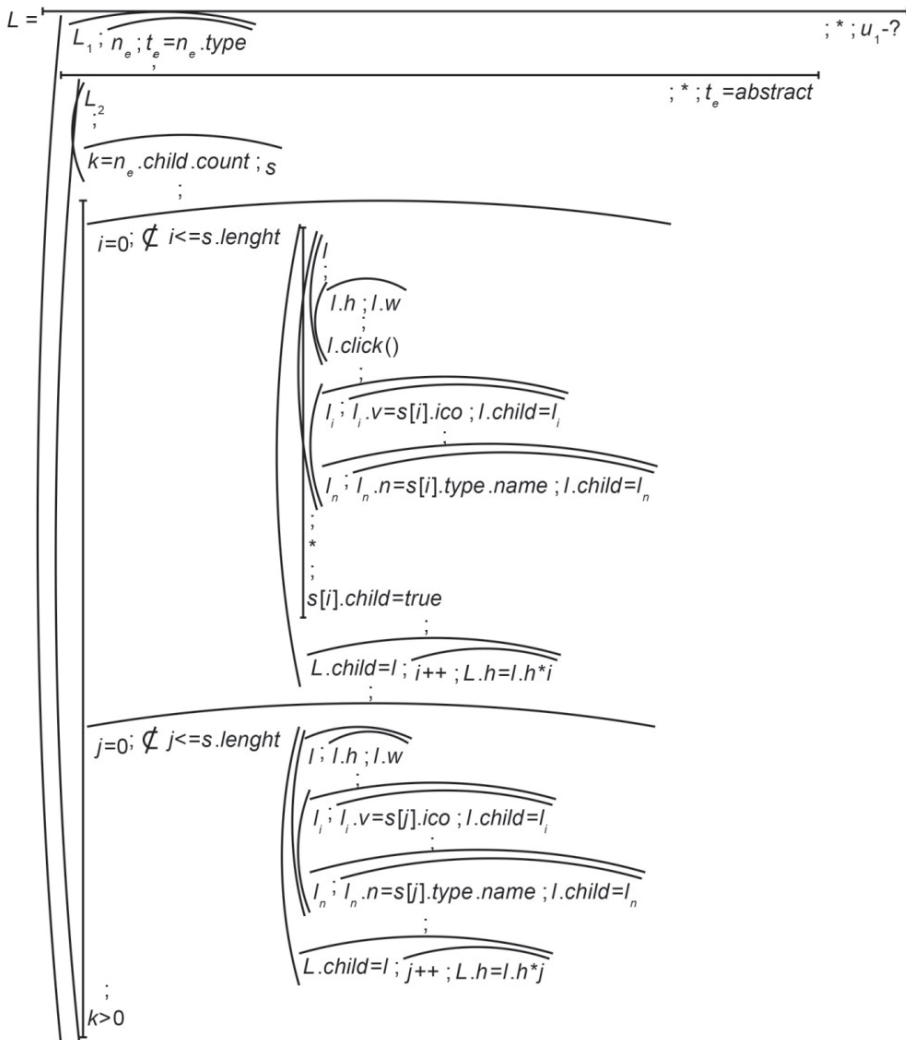


Схема моделі системи типових графічних унітермів для конвертування абстрактного унітерму

Модель побудови системи типових графічних унітермів

Система типових графічних унітермів L формується при виборі користувачем унітерму з розміщених на унітермі-формі (виконання умови u_i , див. наведену нижче формулу алгоритму, для побудови якої застосовано модифіковану алгебру алгоритмів [6–9]). Якщо умова u_i виконується, то за функціональним унітермом L_i визначається унітерм, вибраний користувачем n_e , а також тип вибраного унітерму $t_e = n_e.type$. Зважаючи на те, що конвертувати можна лише унітерм абстрактного типу, доцільно формувати систему L лише при виборі користувачем унітерму абстрактного типу. У зв'язку з тим здійснюється перевірка умови $t_e = abstract$ на виявлення абстрактного унітерму. Якщо умова $t_e = abstract$ виконується, то за функціональним унітермом L_2 визначаються кількість дочірніх унітермів вибраного абстрактного унітерму $k = n_e.count$ і перелік усіх доступних графічних унітермів для конвертації абстрактного. Залежно від кількості дочірніх унітермів k формується система графічних унітермів L . Якщо $k > 0$, тобто обраний абстрактний унітерм містить дочірні унітерми, то формується система L , яка містить графічні унітерми лише контейнерного типу. Для відображення в списку назв типів графічних унітермів у циклі,

поки виконується умова $i \leq s.length$ ($length$ – типова системна властивість [1, 2, 11], яка використовується для обчислення кількості унітермів s системи), генерується поле відображення типу графічного унітерму для конвертації абстрактного. Для реалізації відбору до системи L графічних унітермів лише контейнерного типу здійснюється перевірка умови $s[i].child=true$, тобто перевіряється, чи може графічний унітерм містити дочірні унітерми. Якщо умова $s[i].child=true$ виконується, то генерується поле відображення типу графічного унітерму l із заданими розмірами висоти $l.h$ і ширини $l.w$. Кожному полю l задається подія вибору його користувачем для подальшого конвертування аб-



Фрагмент програмної реалізації моделі

страктного унітерму до вибраного графічного типу. Поле відображення типу графічного унітерму l містить унітерм l_n назви типу графічного унітерму й унітерм графічного позначення унітерму l_i . Згенерованому унітерму l_i задається графічне позначення відповідного графічного унітерму $l_i.v=s[i].ico$ і розміщення його на полі $l.child=l_i$. Унітерму відображення типу графічного унітерму l_n задається назва типу $l_n.n=s[i].type.name$ і проводиться розташування його на полі $l.child=l_n$. Успішно згенероване поле l додається до списку $L.child=l$. З урахуванням кількості визначених контейнерних унітермів формується висота списку $L.h=l.h*i$. Якщо $k=0$, то створюється система L з усіх можливих типів графічних унітермів. Залежно від кількості доступних для конвертації типів графічних унітермів у циклі, поки виконується умова $j \leq s.length$, генерується поле відображення типу графічного унітерму l . Полю l задаються висота $l.h$ і ширина $l.w$ і подія вибору поля користувачем $l.click()$. На полі l розміщуються унітерми відображення назви типу графічного унітерму для конвертації $l.child=l_n$ із заданим значенням $l_n.n=l[j].type.name$, а також графічне позначення відповідного типу $l.child=l_i$ з наступним значенням $l_i.v=s[j].ico$. Успішно згенероване поле відображення типу графічного унітерму l додається до списку $L.child=l$. У цьому випадку висота списку залежатиме від кількості доступних для конвертації обраного абстрактного унітерму типів графічних унітермів $L.h=l.h*j$.

Реалізація моделі конвертації абстрактних унітермів автоматів до типових графічних проілюстрована фрагментом програмного коду мови програмування C# у вигляді методу `private void Convert_Click(object sender, RoutedEventArgs e)`. Даний програмний код реалізує конвертацію абстрактного унітерму до унітерму типу кнопки або типу текстового поля залежно від вибору користувача. Програмно визначається тип унітерму, обраного користувачем, для подальшої конвертації абстрактного унітерму до типового графічного. На основі обраного типу генерується екземпляр елемента цього типу із заданням усіх властивостей, притаманних абстрактному унітерму: розміри, розміщення, колір і т.д. Задаються всі функціональні особливості абстрактного унітерму новому згенерованому. Відбувається перевірка наявності в абстрактного унітерму дочірніх унітермів для розміщення їх на новому графічному унітермі.

```
private void Convert_Click(object sender, RoutedEventArgs e)
{
    removRes();
    if (contr.GetType().Name == «Abstract»)
    {
        FrameworkElement fr = new FrameworkElement();
        SortedList sl = new SortedList();

        Abstract ab = (Abstract)contr;
        if (typStr == «Button»)
```

```
        //Кнопка
    {
        //Створення об'єкта класу типового графічного унітерму
        childbutton = new Button();

        //Передача параметрів абстрактного унітерму
        childbutton.Name = ab.Name;
        childbutton.Content = ab.Name;
        childbutton.Height = ab.Height;
        childbutton.Width = ab.Width;
        Canvas.SetLeft(childbutton, Canvas.GetLeft(ab));
        Canvas.SetTop(childbutton, Canvas.GetTop(ab));

        //Задання назви методу опрацювання події
        childbutton.Click += new RoutedEventHandler(childbutton_Click);
        oblist.Add(childbutton);
        fr = childbutton;

        //Приписування усіх властивостей
        EventL.Add(fr.Name, sl);
    }
    if (typStr == «TextBox»)

        //Текстове поле
    {
        childTextBox = new TextBox();
        childTextBox.Name = ab.Name;
        childTextBox.Height = ab.Height;
        childTextBox.Width = ab.Width;
        Canvas.SetLeft(childTextBox, Canvas.GetLeft(ab));
        Canvas.SetTop(childTextBox, Canvas.GetTop(ab));
        childTextBox.GotFocus += new RoutedEventHandler(childTextBox_Got-
Focus);
        oblist.Add(childTextBox);
        fr = childTextBox;
        EventL.Add(fr.Name, sl);
    }
    //Перевірка і врахування дочірніх унітермів
    switch (ab.Parent.GetType().Name)
    {
        case «Abstract»:
            Abstract a = (Abstract)ab.Parent;
            a.Children.Add(fr);
```

```

a.Children.Remove(ab);
oblist.Remove(ab);
break;
case «Canvas»:
    Canvas c = (Canvas)ab.Parent;
    c.Children.Add(fr);
    c.Children.Remove(ab);
    oblist.Remove(ab);
    break;
case «StackPanel»:
    StackPanel s = (StackPanel)ab.Parent;
    s.Children.Add(fr);
    s.Children.Remove(ab);
    oblist.Remove(ab);
    break;
default:
    break;
} }

```

З вищевикладеного доходимо наступних висновків:

1. Використання створеної моделі забезпечує коректне і безвтратне перетворення абстрактних унітермів у типові графічні.
2. Нові графічні унітерми мають ті самі функціональні особливості, що й абстрактні, задані користувачем на етапі розроблення графічного інтерфейсу.
3. Модель перетворення абстрактних унітермів у типові графічні програмно реалізована й апробована.

1. Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 3.5 с примерами на C# 2008 для профессионалов. / М. Мак-Дональд – М.: ООО И.Д. “Вильямс”, 2008. – 928 с. 2. Трозлсен Э. Язык программирования C# 2010 и платформа .NET 4.0. / Э. Трозлсен. – М.: ООО И.Д. “Вильямс”, 2010. – 1392 с. 3. Овсяк О.В. Модель інформаційної технології опрацювання формул алгоритмів / О.Овсяк // Вісн. Нац. ун-ту “Львівська політехніка”: Комп’ютерні науки та інформаційні технології. – 2011. – № 710. – С. 224–234. 4. Овсяк В. Елементи мови предметних унітермів / Овсяк В., Овсяк О., Василюк А. // Квалілогія книги: зб. наук. пр. – 2009. – № 2(16). – С. 4–9. 5. Овсяк О.В. Інформаційна технологія побудови моделей інтерфейсів інформаційних технологій і систем / О.В. Овсяк // Комп’ютерні технології друкарства: зб. наук. пр. – 2011. – №26. – С.105–114. 6. Ovsyak A. The extended algebra of algorithms width multiconditional elimination / V. Ovsyak, A. Ovsyak // Вісн. Нац. ун-ту “Львівська політехніка”: Комп’ютерні науки та інформаційні технології. – 2010. – № 672. – С. 291–300. 7. Овсяк О. Розширена алгебра алгоритмів і модель зв’язку між класичною і розширеною операцією елімінування / О. Овсяк // Комп’ютерні технології друкарства: зб. наук. пр. – 2010. – №23. – С.45–53. 8. Owskiak A. Rozszerzenie algebry algorytmów / W.Owskiak, A. Owskiak // Pomiar, automatyka, kontrola. – 2010. – №2. – С.184–188. 9. Овсяк О. В. Розширення алгебри алгоритмів аксіомами операції циклів / О. В. Овсяк // Вісн. Нац. ун-ту “Львівська політехніка”: Комп’ютерні системи проектування. Теорія і практика. – 2010. – № 685. – С. 12–20. 10. Ovsyak A. The extended algebra of algorithms with additional cycle elimination axioms. / A. Ovsyak, V. Ovsyak // Conference “Intelligent Information and Engineering Systems” (INFOS 2011), September 19–23, 2011, Polańczyk, Poland. – P. 23–34. 11. Шилд Г. C# 4.0: полное руководство / Г. Шилд. – М.: ООО И.Д. “Вильямс”, 2011. – 1056 с.

МОДЕЛЬ КОНВЕРТАЦИИ АБСТРАКТНЫХ К ТИПОВЫМ ГРАФИЧЕСКИМ УНИТЕРМОВ АВТОМАТОВ

Средствами алгебры алгоритмов описано созданную модель построения системы типовых графических унитермов, предназначенных для конвертации абстрактного унитерма.

MODEL RECOGNITION ABSTRACT TO THE DEFAULT GRAPHICS UNITERMS AUTOMATS

Means algebra algorithms described created a model of the system default image unitermiv designed to convert abstract uniterma.

Стаття надійшла 01.10.2012

УДК 655.3.066.252

Р. Б. Стахів

Українська академія друкарства

МЕТОДИ ОПИСУ ГРАФІЧНИХ СКЛАДОВИХ ЗАСОБІВ ЗАХИСТУ ПРОДУКЦІЇ МАСОВОГО ПОПИТУ

Аналізуються теоретичні засоби, що описують методи захисту етикетки від підміни чи підробки.

Захист продукції, засоби захисту, графічні складові, етикетка, підміна, підробка

Для проведення коректного аналізу теоретичних засобів, що описують методи захисту етикетки від підміни чи підробки, введемо обмеження на об'єкт дослідження, які враховуватимемо при формулюванні теретичних положень. На відміну від існуючих уявлень про етикетку впровадимо особисте визначення. Під етикеткою розуміємо довільний носій інформації, задруковуваної на ньому, що містить дані, які дозволяють повністю ідентифікувати продукт виробництва.

Етикетка може бути привабливою і непривабливою. До останніх відносяться етикетки, які за допомогою кольорів, що формують власний сюжет, виділяють місця розміщення інформації для споживачів, але до того ж вони відтворені таким чином, що зчитування інформації є доволі важким. Це спричиняється переважно підбором розмірів шрифту, вибором неконтрастного фону для роздрукованої інформації, розміщенням інформації на складнодоступних місцях пакування тощо.

Оскільки інформація, розміщена на етикетці, використовується для ідентифікації продукту, то вона повинна бути захищена від підробок. Згідно з усталеним визначенням, будь-який носій інформації, що має засоби захисту, етикетку можна віднести до класу документів [4]. У цьому випадку її захист як однієї з різновидностей документів слід розглядати в рамках проблеми захисту ділових паперів.