

УДК 004

О. В. Овсяк

Українська академія друкарства

Львівська філія Київського національного університету культури і мистецтв

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИБОРУ ТА ОБЧИСЛЕННЯ
РОЗМІРІВ ОПЕРАЦІЇ ЕЛІМІНУВАННЯ**

Описана декомпозиція інформаційної технології комп'ютерного опрацювання операції елімінування. Наведено математичні моделі опису змінних, вибору та обчислення розмірів операції елімінування.

Інформаційна технологія, модель, декомпозиція, елімінування, алгебра алгоритмів

В алгебрі алгоритмів [5] є операція елімінування, призначена для опису їх розгалужень. Знаку операції елімінування серед відомих позначень немає. Його можна створити відомими комп'ютерними системами, наприклад Word, Coral Draw та іншими. Їх використання підвищує рівень автоматизації набору і редагування формул алгоритмів. Але довжина і ширина знаку операції елімінування залежить від розмірів знаків, над якими вона виконується. Постійна зміна розмірів знаку операції елімінування робить використання універсальних комп'ютерних систем неефективним. Через те було створено спеціалізовані редактори «Модал» [1] і «Абстрактал» [2]. Застосування спеціалізованих комп'ютерних систем порівняно з універсальними є більш ефективним. Однак цими системами використовуються спеціальні та жорсткі формати даних, що при внесенні змін спричиняє необхідність побудови складних алгоритмів і програм. У зв'язку з тим у статті розглянуто побудову математичних моделей вибору і обчислення розмірів операції елімінування, які використовують її *xml*-опис. Цей формат відкриває можливість застосування розмаїтих типових напрацювань, які суттєво підвищують гнучкість створюваних засобів.

МОДЕЛЬ ДЕКОМПОЗИЦІЇ ПІДСИСТЕМИ ЕЛІМІНУВАННЯ

Для ідентифікації підсистеми застосуємо знак @. Нехай @E є позначенням підсистеми елімінування. Модель підсистеми (@E) використовує змінні і функційні унітерми [5] підсистеми @T [3], що запишемо як @E:@T, де : — знак належності (наслідування) унітермів підсистеми, записаної справа до підсистеми, записаної зліва від цього знаку. У зв'язку з тим, що підсистема елімінування має бути доступною з інших підсистем, то задаємо до неї загальний метод доступу (*pu*), який є типовим і програмно реалізується *public* [4, 6]. Утворюємо модель цієї підсистеми унітермами — змінними (Z), призначеними для збереження тимчасових даних, функційними унітермами задання початкових значень змінним (E()), обчислення розмірів унітермів і розділювача між ними (Cs()), ідентифікації вибору елімінування (Cfc()), деселекції вибору (Des()), обчислення розмірів і рисування рамки вибору знаку операції (Dra()), рисування тесту унітерма (DraTex()), обчислення довжини (TexLen()) і висоти (TexHei()) текстів унітермів та видалення (VydEli()) операції елімінування. Усі унітерми є необхідними, тому для їх опису потрібно застосувати операцію секвентування. Але місце взаємного роз-

ташування унітермів не має значення, тому розділювачем унітермів вибираємо кому, яка в алгебрі алгоритмів використовується для опису властивості комутативності операцій. З огляду на це запис моделі декомпозиції підсистеми на унітерми матиме такий вигляд:

$$\underline{pu} \ @E:@T = \left(\overbrace{\left(\overbrace{Z, E(), Cs(), Cfc(), Des(), Dra(),} \right)} \right) \overbrace{\left(\overbrace{DraTex(), TexHei(), TexLen(), FormTex(), VydEli()} \right)}$$

МОДЕЛІ СКЛАДОВИХ ДЕКОМПОЗИЦІЇ

Модель змінних (Z)

Операція елімінування має горизонтальну (*Hor*) і вертикальну (*Ver*) орієнтації [5]. Для їхнього задання вводимо загальнодоступну (pu) секвентну область *Ori* типу enu, який реалізується еnum стандартного зразка [4, 6]:

$$\underline{pu} \ Ori \in \underline{enu} = \overbrace{Hor; Ver}$$

Операція елімінування виконується над трьома унітермами. Нехай значення унітермів приписуються загальнодоступним змінним pu $tA \in @T$, pu $tB \in @T$ та pu $con \in @T$. Вводимо загальнодоступну (pu) змінну *ori* для збереження значення орієнтації: pu *oriOri*. Для задання кольору знаку операції вводимо загальнодоступну змінну pu $sD \in @Bru.Gol$, де *Bru* — підсистема, що реалізується стандартним класом Brush [4, 6] і *Gol* — назва кольору, який реалізується відомим кольором Gold [4, 6]. Змінна pu $fI \in @Bru.Red$ вводиться для виділення операції елімінування кольором *Red*, що реалізується відомим кольором Red [4, 6].

Нехай збережені значення координат є величинами змінної *po* типу підсистеми *Poi*, яка реалізується відомою стандартною підсистемою Point [4, 6], що опишемо як pu $po \in @Poi$.

Збереження кількості знаків операцій у змінній *kF* типу *Int* реалізується відомим стандартним типом int [4, 6]: $kF \in @Int$.

Модель функційного унітерма $E()$

Алгоритм є загальнодоступним і описується формулою

$$\underline{pu} \ E() = \left(\begin{array}{l} tA=\$ \\ ; \\ tB=\$ \\ ; \\ \left(\begin{array}{l} con=\$ \\ ; \\ ori=Ori.Hor, \end{array} \right) \end{array} \right)$$

якою описується задання початкових значень змінним *tA*, *tB*, *con* і *ori*, де $\$$ — початкове значення змінних типу підсистеми.

Обчислення розмірів виразу елімінування ($Cs()$)

Геометричні розміри знаку операції елімінування залежать від розмірів виразу, який знаходиться під ним. Вираз утворено трьома унітермами і розділювачем між

ними. Для доступності функційного унітерма з інших підсистем робимо його загальнодоступним з можливістю в даній підсистемі переозначення (ov) дефініції останнього, яка описана в наслідуваній підсистемі. Нехай входними параметрами є $dv \in @DraV$ типу підсистеми DraV, реалізованої відомим класом DravingVisual [4, 6] і $f \in @Siz$ типу підсистеми Siz, яка реалізована відомим класом Size [4, 6], а сам функційний унітерм описується формулою

$$pu\ ov\ Cs(dv \in @DraV, f \in @Siz) =$$

```

sepSiz ∈ @Siz = @Siz(TexLen(;), TexHei(;))
;
wid = sepSiz..Wid*2 + f.Hei*3
;
hei = sepSiz.Hei
;
(tA.Cs(dv, f); *; (tA≠$)-?)
;
wid = wid + tA.wid
;
hei = tA.hei; *; (hei < tA.hei)-?
;
(tB.Cs(dv, f); *; (tB≠$)-?)
;
wid = wid + tB.wid
;
hei = tB.hei; *; (hei < tB.hei)-?
;
(con.Cs(dv, f); *; (con≠$)-?)
;
wid = con.wid
;
hei = con.hei; *; (hei < con.hei)-?
;
hei = hei + Mat.Sqrt(wid)/2 + 2
;
W
;
(ori = Ori.Hor)-?

```

де $sepSiz \in @Siz = @Siz(TexLen(;), TexHei(;))$ — унітерм обчислення довжини TexLen(;) і TexHei(;) висоти розділювача (;) унітермів операції елімінування і приписання їх змінній sepSiz типу Siz; $wid = sepSiz..Wid*2 + f.Hei*3$ — збільшення довжини виразу під знаком операції елімінування з урахуванням довжин двох розділювачів унітермів і розміру висоти вибраного кегля $f.Hei*3$; $hei = sepSiz.Hei$ — приписування змінній висоти виразу hei значення висоти розділювача sepSiz.Hei; $(tA \neq \$)-?$ — опис перевірки наявності першого зліва унітерму (tA) операції елімінування; tA.Cs(dv, f) — вибір функційного унітерма Cs(dv, f) підсистеми унітерм (U) та обчислення довжини першого унітерма; $wid = wid + tA.wid$ — збільшення загальної довжини виразу під знаком операції елімінування з урахуванням довжини першого унітерму; $hei = tA.hei; *$;

$(hei < tA.hei)$ -? — порівняння поточного значення висоти виразу з висотою першого унітерма; $hei = tA.hei$ — приписування змінній поточної висоти виразу значення висоти першого унітерма (у тому випадку, коли значення поточної висоти менше за значення висоти першого унітерма); $(tB \neq \$)$ -? — перевірка наявності другого зліва унітерму (tB) операції елімінування; $tB.Cs(dv, f)$ — вибір функційного унітерма $Cs(dv, f)$ підсистеми унітерм (U) та обчислення довжини другого унітерма tB ; $wid = wid + tB.wid$ — збільшення загальної довжини виразу під знаком операції елімінування з урахуванням довжини другого унітерму; $(hei < tB.hei)$ -? — порівняння поточного значення висоти виразу з висотою другого унітерма; $hei = tB.hei$ — приписування змінній поточної висоти виразу значення висоти другого унітерма (у тому випадку, коли значення поточної висоти менше за значення висоти другого унітерма); $(con \neq \$)$ -? — перевірка наявності умовного унітерма; $con.Cs(dv, f)$ — вибір функційного унітерма $Cs(dv, f)$ підсистеми унітерм (U) та обчислення довжини умовного унітерма con ; $wid = con.wid$ — збільшення загальної довжини виразу під знаком операції елімінування з урахуванням довжини умовного унітерма; $hei < con.hei$ -? — порівняння поточного значення висоти виразу з висотою умовного унітерма; $hei = con.hei$ — приписування змінній поточної висоти виразу значення висоти умовного унітерма (у тому випадку, коли значення поточної висоти менше за значення висоти умовного унітерма); $hei = hei + \text{Mat.Sqr}(wid)/2 + 2$ — обчислення висоти знаку операції елімінування з урахуванням співвідношення між довжиною і висотою; $(ori = Ori.Hor)$ -? — перевірка значення змінної орієнтації операції елімінування; W — формула, яка описує обчислення знаку операції елімінування для її вертикальної орієнтації і містить унітерми, аналогічні унітермам горизонтальної орієнтації операції.

$$\begin{aligned}
 W = & \left(\begin{array}{l}
 \text{wid} = \text{sepSiz}.. \text{wid} \\
 ; \\
 \text{hei} = \text{sepSiz}.. \text{Hei} * 2 + f. \text{Hei} + 16 \\
 ; \\
 \text{tA.Cs}(dv, f); *; (tA \neq \$)-? \\
 ; \\
 \text{wid} = \text{tA.wid}; *; (\text{wid} < \text{tA.wid})-? \\
 ; \\
 \text{hei} = \text{hei} + \text{tA.hei} \\
 ; \\
 \text{tB.Cs}(dv, f); *; (tB \neq \$)-? \\
 ; \\
 \text{wid} = \text{tB.wid}; *; (\text{wid} < \text{tB.wid})-? \\
 ; \\
 \text{hei} = \text{hei} + \text{tB.hei} \\
 ; \\
 \text{con.Cs}(dv, f); *; (con \neq \$)-? \\
 ; \\
 \text{wid} = \text{con.wid}; *; (\text{wid} < \text{con.wid})-? \\
 ; \\
 \text{Hei} = \text{hei} + \text{con.hei} \\
 ; \\
 \text{wid} = \text{wid} + \text{Mat.Sqr}(\text{hei})/2 + 2
 \end{array} \right)
 \end{aligned}$$

Модель встановлення вибору операції елімінування (*Cfc()*)

Алгоритм вибирається з інших підсистем, наприклад, підсистем секвентування і паралелення, тому задаємо до нього загальний доступ. Функційний унітерм використовує підсистему терм для опрацювання унітермів операції елімінування, тому потрібно, щоб він мав вихідну змінну (*te*) типу підсистеми терм (*@T*). Вибір операції елімінування здійснюється встановленням курсора в прямокутник операції та фіксацією координат курсора після натискання лівою клавішею мишки. Знаходження координат курсора в області прямокутника операції елімінування означає вибір операції елімінування. У зв'язку з тим для функційного унітерма вибору необхідні такі вхідні змінні: *mf* — головної підсистеми *Mf*, для доступу до її унітермів; *f* — типу підсистеми *Siz*, *x-y-mX-mY-moX-moY* (*x* і *y* — поточних значень абсциси та ординати розташування операції; *mX* і *mY* — значень абсциси й ординати відступів робочого стола від краю форми; *moX* і *moY* — значень абсциси й ординати курсора — типу стандартної підсистеми *Dou*, яка реалізована відомим типом *Double* [4, 6]. Сам алгоритм описується формулою

$$\underline{pv\ ov} (te \in @T) Cfc(mf \in @Mf, f \in @Siz, x-y-mX-mY-moX-moY \in @Dou) =$$

```

sT ∈ @T
;
sepSiz = @Siz( TexLen(sepS), TexHei(sepS) )
;
xT ∈ @Dou = x
;
y ∈ @Dou = y
;
F; Q; (ori=OriHor)-?
;
rec ∈ @Rec(x-f.Hei/4+4, y-f.Hei/8+2, wid+f.Hei/2+4,
           hei+f.Hei/4+4);
mf.c-cD=tru; te=$.; (sel=rec.Cont(moX, moY)-?)
mf.sel cD=fal;
te=thi.

```

де $sT \in @T$ — введення змінної типу підсистеми *T*; $sepSiz = @Siz(TexLen(sepS), TexHei(sepS))$ — обчислення довжини і ширини розділювача унітермів і приписування їх значень змінній *sepSiz*; $xT \in @Dou = x$ — введення змінної для збереження значення абсциси розташування знаку операції елімінування створення змінної *xT* типу стандартної підсистеми *Dou* і приписування їй значення змінної *x*, $y \in @Dou = y$ — введення змінної для збереження значення ординати розташування знаку операції елімінування і приписування їй значення змінної *y*; $(ori=OriHor)-?$ — порівняння значення змінної зі значенням горизонтальної орієнтації операції елімінування *OriHor*; *F* і *Q* — формули опису обчислення розмірів операції горизонтальної і вертикальної орієнтації; $rec \in @Rec(x-f.Hei/4+4, y-f.Hei/8+2, wid+f.Hei/2+4, hei+f.Hei/4+4)$ — обчислення розмірів прямокутної області операції, $(sel=rec.Cont(moX, moY)-?)$ — із застосуванням унітерма *Cont()*, який реалізується стандартним методом *Contains()* [4, 6], перевірка попадання *Cont(moX, moY)* у прямокутну область операції елімінування *rec* координат курсора

moX і moY , що зафіксовані натисканням клавіші мишки та приписування змінній sel результату порівняння; $mf.c-cD=tru$ — приписування змінній $c-cD$ головної підсистеми стандартного значення tru ; $te=\$$. — приписування вихідній змінній te значення $\$$ і завершення (.) алгоритму, $mf.sel_cD=fal$ — приписування (=) стандартного значення fal змінній sel_cD головної підсистеми; $te=thi$. — приписування змінній стандартного значення thi , яке реалізується $this$ [4, 6] і завершення опису алгоритму.

$$\begin{aligned}
 F = & \left(\begin{array}{l}
 \underline{sH} = \underline{Mat.Sqr}(\underline{wid})/2 + 2 \\
 ; \\
 \underline{xT} = \underline{xT} + \underline{f.Hei}/2 \\
 ; \\
 \underline{yT} = \underline{sH} \\
 ; \\
 \underline{te} = \underline{sT}; \underline{xT} = \underline{xT} + \underline{tA.wid}; \underline{u}_1\text{-?} \\
 ; \\
 \underline{xT} = \underline{xT} + \underline{f.Hei} + \underline{sepSiz.Wid} \\
 ; \\
 \underline{te} = \underline{sT}; \underline{xT} = \underline{xT} + \underline{tA.wid}; \underline{u}_2\text{-?} \\
 ; \\
 \underline{xT} = \underline{xT} + \underline{f.Hei} + \underline{sepSiz.Wid} \\
 ; \\
 \underline{te} = \underline{sT}; \underline{*}; \underline{u}_3\text{-?}
 \end{array} \right) ; \underline{*}; (\underline{tA} \neq \underline{\$})\text{-?} \\
 & ; \underline{*}; (\underline{tB} \neq \underline{\$})\text{-?} \\
 & ; \underline{*}; (\underline{con} \neq \underline{\$})\text{-?}
 \end{aligned}$$

У формулі F маємо: $sH = \underline{Mat.Sqr}(\underline{wid})/2 + 2$ — обчислення висоти sH знаку операції елімінування як поділений на два корінь квадратний $\underline{Sqr}()$ з його довжини \underline{wid} плюс два пікселі, для чого використовується стандартний функційний унітерм $\underline{Sqr}()$ стандартної підсистеми \underline{Mat} , які реалізуються відомим методом $\underline{Sqr}()$ підсистеми \underline{Math} [4, 6], $\underline{xT} = \underline{xT} + \underline{f.Hei}/2$ — обчислення ординати з урахуванням розміру кегля унітермів; $\underline{yT} = \underline{sH}$ — приписування ординаті значення змінної sH ; $(\underline{tA} \neq \underline{\$})\text{-?}$ — перевірка наявності першого зліва унітерму; $\underline{u}_1\text{-?}$ — умовний унітерм, який описується таким виразом:

$$\underline{u}_1 = (\underline{selT} = \underline{tA.Cfc}(\underline{mf} \in @\underline{Mf}, \underline{f} \in @\underline{Siz}, \underline{x-y-mX-mY-moX-moY} \in @\underline{Dou}) \neq \underline{\$}),$$

де $\underline{te} = \underline{sT}$ — приписування вихідній змінній te значення змінної sT , $\underline{xT} = \underline{xT} + \underline{tA.wid}$ — обчислення абсциси знаку xT операції з урахуванням довжини \underline{wid} унітерму \underline{tA} , $\underline{xT} = \underline{xT} + \underline{f.Hei} + \underline{sepSiz.Wid}$ — обчислення абсциси з урахуванням розмірів кеглю $\underline{f.Hei}$ і розділювача $\underline{sepSiz.Wid}$ унітермів, $(\underline{tB} \neq \underline{\$})\text{-?}$ — перевірка наявності другого унітерма \underline{tB} , $\underline{u}_2\text{-?}$ — умовний унітерм, який описується виразом $\underline{u}_2 = (\underline{selT} = \underline{tB.Cfc}(\underline{mf} \in @\underline{Mf}, \underline{f} \in @\underline{Siz}, \underline{x-y-mX-mY-moX-moY} \in @\underline{Dou}) \neq \underline{\$}); (\underline{con} \neq \underline{\$})\text{-?}$ — перевірка наявності умовного унітерма; $\underline{u}_3\text{-?}$ — умовний унітерм, який описується виразом: $\underline{u}_3 = (\underline{selT} = \underline{cond.Cfc}(\underline{mf} \in @\underline{Mf}, \underline{f} \in @\underline{Siz}, \underline{x-y-mX-mY-moX-moY} \in @\underline{Dou}) \neq \underline{\$})$.

Вибір операції елімінування з вертикальною орієнтацією описується формулою Q , аналогічною формулі W .

$$Q = \left(\begin{array}{l} sW = \text{Mat.Sqr}(hei) / 2 + 2 \\ ; \\ xT = xT + sW \\ ; \\ yT = yT + f.Hei / 2 \\ ; \\ \text{---}; (tA \neq \$) - ? \\ te = sT; yT = yT + tA.hei; u_1 - ? \\ ; \\ yT = yT + \delta + sepSiz.Hei \\ ; \\ \text{---}; (tB \neq \$) - ? \\ te = sT; yT = yT + tB.hei; u_2 - ? \\ ; \\ yT = yT + \delta + sepSiz.Hei \\ ; \\ \text{---}; (con \neq \$) - ? \\ te = sT; *; u_3 - ? \end{array} \right.$$

ФРАГМЕНТ ПРОГРАМНОГО КОДУ

Запрограмований мовою C# [4, 6] код функційного унітерма вибору операції елімінування є таким:

```
public override Term CheckForClick(MainForm mf, Size
f, double x, double y, double marginX, double marginY, double
mouseX, double mouseY)
{
    Size separatorSize = new Size(GetTextLength(<<;>>),
GetTextHeight(<<;>>));
    Term selectedTerm;
    double xTmp = x;
    double yTmp = y;
    if (orientation == Orientation.Horizontal)
    {
        double symbolHeight = (Math.Sqrt(width) / 2) + 2;
        xTmp += (f.Height / 2);
        yTmp += symbolHeight;
        if (termA != null)
        {
            if ((selectedTerm = termA.CheckForClick(mf, f,
xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
                return selectedTerm;
            xTmp += termA.width;
        }
        xTmp += f.Height / 2);
        xTmp += separatorSize.Width;
        xTmp += (f.Height / 2);
        if (termB != null)
```

```
{
    if ((selectedTerm = termB.CheckForClick(mf, f,
xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
        return selectedTerm;

    xTmp += termB.width;
}
xTmp += (f.Height / 2);
xTmp += separatorSize.Width;
xTmp += (f.Height / 2);
if (condition != null)
{
    if ((selectedTerm = condition.CheckForClick(mf,
f, xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
        return selectedTerm;
}
}
else
{
    double symbolWidth = (Math.Sqrt(height) / 2) + 2;
    xTmp += symbolWidth;
    yTmp += (f.Height / 2);
    if (termA != null)
    {
        if ((selectedTerm = termA.CheckForClick(mf, f,
xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
            return selectedTerm;
        yTmp += termA.height;
    }
    yTmp += 4;
    yTmp += separatorSize.Height;
    yTmp += 4;
    if (termB != null)
    {
        if ((selectedTerm = termB.CheckForClick(mf, f,
xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
            return selectedTerm;
        yTmp += termB.height;
    }
    yTmp += 4;
    yTmp += separatorSize.Height;
    yTmp += 4;
    if (condition != null)
    {
        if ((selectedTerm = condition.CheckForClick(mf,
f, xTmp, yTmp, marginX, marginY, mouseX, mouseY)) != null)
            return selectedTerm;
    }
}
```

```

    }
}
Rect rect = new Rect(x - ((f.Height / 4) + 4),
y - ((f.Height / 8) + 2),
width + (((f.Height / 4) * 2) + 4),
height + ((f.Size*/f.Height / 8) * 2) + 4));
if (selected = rect.Contains(mouseX, mouseY))
{
    mf.cok_Na_canvasDraw = true;
    mf.selected_Uniter = false;
    return this;
}
return null;
}
}

```

З вищевикладеного випливає:

1. Виконаною декомпозицією системи елімінування на підсистеми спрощено складність створення математичної моделі системи.
2. Розроблені моделі функційних унітермів описують обчислення розмірів операції та ідентифікацію вибору операції елімінування.
3. Виконаною верифікацією підтверджено коректність побудованих моделей.

1. Бритковський В.М. *Моделювання редактора формул секвенційних алгоритмів: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 01.05.02 «Математичне моделювання та обчислювальні методи»* / Бритковський В.М. — Львів, 2003. — 18 с. 2. Василюк А. С. *Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 01.05.02 «Математичне та програмне забезпечення обчислювальних машин і систем»* / Василюк А. С. — Львів, 2008. — 20 с. 3. Овсяк О. *Класи інформаційної системи генерування коду /О. Овсяк // Вісн. Терноп. нац. техн. ун-ту імені Івана Пулюя. — 2010. — № 1 — С. 171 — 176.* 4. MacDonald M. *Windows presentation foundation в .NET 3.5 с прикладами на C# 2008* /М. Мак-Дональд. — М., СПб., К.: Apress, 2008. — 922 с. 5. Owsiak W. *Rozszerzenie algebry algorytmów* / W. Owsiak, A. Owsiak // *Pomiary, automatyka, kontrola.* — 2010. — № 2, — S. 184 — 188. 6. Petzold C. *Programowanie Windows w języku C#* /C. Petzold. — Warszawa: RM, 2002. — 1161 s.

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ВЫБОРА И ВЫЧИСЛЕНИЯ РАЗМЕРОВ ОПЕРАЦИИ ЕЛИМИНИРОВАНИЯ

Описана декомпозиция информационной технологии компьютерной обработки операции элиминирования. Поданы математические модели описания сменных, выбора и вычисления размеров операции элиминирования.

INFORMATION TECHNOLOGY SELECTION AND CALCULATION DIMENSIONS OF OPERATION ELIMINATION

We describe a decomposition of information technology computer processing operation elimination. Given mathematical model selection and calculation dimensions of operation elimination.

Стаття надійшла 07.02.11

УДК 004.5 : 655.1/.3

Є. В. Пономаренко*Харківський національний економічний університет***МЕТОДИКА СТВОРЕННЯ ЕФЕКТИВНОГО ІНТЕРФЕЙСУ
WEB-САЙТУ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА**

Порушується питання створення ефективного інтерфейсу web-сайту поліграфічного підприємства на основі запропонованої методики. Пропонована методика має становити концептуальну основу процесу створення сайтів підприємств поліграфічної галузі.

Інтерфейс сайту, створення web-сайту, навігація по сайту, ефективність інтерфейсу

Одним з найважливіших завдань для замовників поліграфічного підприємства є можливість цілодобово й оперативно одержувати свіжу інформацію про спектр видавничих послуг. При цьому велике значення має інтерактивність, тобто здатність діалогового спілкування з видавцями. Така можливість найефективніше забезпечується за допомогою web-сайтів. Разом з тим Інтернет створює зручність спілкування з клієнтами, що знаходяться в будь-якому куточку земної кулі. Співробітники й партнери фірми завжди отримуватимуть необхідні дані, інструкції, документи або будь-яку іншу службову інформацію, де б вони не перебували. Інтернет дає змогу обновляти відомості на сайті настільки часто, наскільки в цьому є потреба.

Розглянута нами література містить безліч порад щодо створення інтерфейсу сайту, що дозволить відвідувачеві легко орієнтуватися в його ресурсах. У дослідженнях [1–4] окреслено основні принципи юзабіліті відносно створення сайтів, пропонуються рекомендації з використання компонент інтерфейсу, оформлення елементів навігації по сайту. Однак в усіх цих публікаціях відсутня цілісна методика створення ефективного інтерфейсу сайту поліграфічного підприємства.

Метою даної статті є проектування методики розроблення ефективного інтерфейсу web-сайту підприємства.

Структурно-логічна модель предметної області наукового дослідження подана на рис. 1.

Для повноцінної роботи будь-якого сайту повинен існувати не тільки зв'язок web-майстер — відвідувач, але й зворотний (відвідувач — web-майстер). Наявність можливості зв'язатися з розробниками ресурсу надто важлива перелюдям для комерційних проектів, оскільки для них відвідувач ототожнюється з потенційним клієнтом.

Дані методичні рекомендації призначені для застосування при створенні web-сайту поліграфічного підприємства й спрямовані на максимальну взаємодію із замовниками (клієнтами, партнерами).