

фону. Отже, при якісному забезпеченні критеріїв вищих рівнів можна забезпечити зручність читання і сприйняття тексту в електронних виданнях.

За допомогою такого моделювання критеріїв дизайнеру буде простіше визначити пріоритети під час розробки текстового наповнення видання для різних носіїв. На основі таких моделей можна здійснити їх подальшу оптимізацію для створення відповідних програм, за допомогою яких автоматично здійснювати проектування оформлення тексту в електронних виданнях.

1. Каров П. Шрифтовые технологии. Описание и инструментарий / П. Каров. — М., 2001.
2. Клеппер М. Л. Практическое руководство по цифровой печати / М. Л. Клеппер. — М., 2003.
3. Кудрявцев А. И. Шрифт: история, теория, практика / А. И. Кудрявцев. — М., 2002.
4. Лямець В. І. Системний аналіз: вступний курс. — 2-е вид., переробл та допов. / В. І. Лямець, А. Д. Тевяшев. — Х. : ХНУРЕ, 2004.
5. Сеньківський В. М. Модель ієрархії критеріїв якості книжкових видань / В. М. Сеньківський // Наукові записки УАД. — 2007. — Вип. 11. — С. 73–80.
6. Токарь О. В. Удобочитаемость современных текстовых шрифтов. — Мн. : Современная школа, 2007.
7. Феличи Дж. Типографика: шрифт, верстка, дизайн / Дж. Феличи. — СПб., 2004.

МОДЕЛЬ КРИТЕРИЕВ УДОБСТВА ЧТЕНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ИЗДАНИЯХ

Из перечня проанализированных критериев, влияющих на удобство чтения текста в электронных изданиях, разработан граф взаимосвязей между критериями, которые иерархически упорядочены по приоритетности влияния на процесс моделирования оформления текста в электронных изданиях.

MODEL CRITERIA READABILITY OF TEXT INFORMATION IN ELECTRONIC PUBLICATIONS

Based on the analyzed criteria that affect the readability of text in electronic editions designed graph relationships between criteria that are hierarchically organized by prioritetnistly influence on the process modeling design text electronic editions.

Стаття надійшла 29.09.2011

УДК 004

О. В. Овсяк

Українська академія друкарства,

Львівська філія Київського національного університету культури і мистецтв

РЕКУРЕНТНО-ДЕКОМПОЗИЦІЙНА МЕТОДОЛОГІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

Описується створена рекурентно-декомпозиційна методологія побудови моделей інформаційних технологій і систем. На її основі побудовано структурну і функціональну рекурентно-декомпозиційні моделі інформаційної технології генерування програмного коду з формул алгоритмів.

Алгебра алгоритмів, метод, декомпозиція, модель, рекуренція

Сучасні інформаційні технології і системи мають графічні інтерфейси і реалізують достатньо багато різноманітних функцій. Їх проектування є складним науково-технічним завданням. Проектування нових інформаційних технологій і систем, для зменшення його складності, виконується декомпозиційним методом. Для розбиття моделі використовуються різноманітні критерії, наприклад, функціонального призначення, модульності тощо.

Методом є прийом або система прийомів, що застосовуються у будь-якій галузі діяльності (науці, виробництві тощо) [1]. Методологією є сукупність методів дослідження, що застосовуються у будь-якій науці відповідно до специфіки об'єкта її пізнання [1]. Рекурентною формулою (*рекурентним співвідношенням*) є формула, яка описує обчислення всіх членів послідовності, якщо задано декілька її перших членів [2]. Декомпозицією є розбиття складної системи на підсистеми [4]. Розрізняють різні типи декомпозиції, наприклад, структурну, функціональну [4].

У загальному випадку структурна чи функціональна декомпозиція є багаторівневою і впорядкованою. Використовувана методологія декомпозиції базується на інтуїтивному описі впорядкованості. На основі розширеної алгебри алгоритмів впорядкованість описується формально. Моделі інформаційних технологій і систем можуть бути побудовані на основі рекурентно-декомпозиційної методології, формулювання якої і застосування до проектування описано нижче.

Суть рекурентно-декомпозиційної методології в аналітичному описі впорядкованої багаторівневої декомпозиції інформаційних технологій і систем на складові – унітерми з кінцевим розбиттям на тривіальні та функційні унітерми.

Проектування рекурентно-декомпозиційної структури. Нехай потрібно створити рекурентно-декомпозиційну структуру унітерму F . Для виконання декомпозиції потрібно вибрати ознаки, за якими вона проводитиметься. Наприклад, такими ознаками можуть бути виконувані функції, внутрішня специфіка створюваної технології чи системи, вимоги й обмеження тощо. Нехай ознака або декілька ознак є вибраними. За цими ознаками спочатку (на першому рівні) декомпозуємо унітерм F на «підунітерми» $P_0, P_1, \dots, P_{p-1}$. Очевидно, що в F входять усі p унітермів від P_0 до P_{p-1} . Обов'язкову наявність усіх унітермів описуємо операцією секвентування з розділювачем між унітермами двох крапок. У разі, якщо має значення черговість (взаємне розташування) унітермів P_i та P_j , то у секвентуванні замість двох крапок застосовуємо крапку з комою. Якщо ж черговість немає значення, то розділювачем унітермів операції секвентування вибираємо кому. Між секвенцією унітермів ставимо знак рівності.

За декомпозицією першого рівня виконуємо декомпозицію другого рівня, яка полягає у розбитті унітермів $P_0, P_1, \dots, P_{p-1}$ на підунітерми. Наприклад, унітерм P_0 декомпозовано на унітерми $Q_0, Q_1, \dots, Q_{q-1}$, а унітерм P_{p-1} на унітерми $R_0, R_1, \dots, R_{r-1}$. Аналогічно розбиваємо всі решти унітермів. Кожному з унітермів $P_0, P_1, \dots, P_{p-1}$ приписуємо секвенцію його «підунітермів»

з відповідними розділювачами. Серед унітермів $P_0, P_1, \dots, P_{p-1}$ можливі такі унітерми, які не підлягають або декомпозицію яких недоцільно виконувати.

Далі виконується декомпозиція на другому, третьому і подальших рівнях. Завершується виконання декомпозиції предметними і функційними унітермами.

Загальний вигляд рекурентної моделі структурної декомпозиції описано формулою (1)

$$\begin{aligned}
 F = \left(\begin{array}{l} P_0 = \left(\begin{array}{l} Q_0 = \dots \left(\begin{array}{l} S_0 = \overline{U_0: U_1: \dots U_{s-1}} \\ \vdots \\ S_1 = \overline{L_0: L_1: \dots L_{s-1}} \\ \vdots \\ S_{s-1} = \overline{K_0: K_1: \dots K_{s-1}} \end{array} \right) \\ \vdots \\ Q_1 = \left(\begin{array}{l} E_0 = \overline{A_0: A_1: \dots A_{s-1}} \\ \vdots \\ E_1 = \overline{H_0: H_1: \dots H_{s-1}} \\ \vdots \\ E_{s-1} = \overline{B_0: B_1: \dots B_{s-1}} \end{array} \right) \\ \vdots \\ Q_{s-1} = \left(\begin{array}{l} D_0 = \overline{C_0: C_1: \dots C_{s-1}} \\ \vdots \\ D_1 = \overline{G_0: G_1: \dots G_{s-1}} \\ \vdots \\ D_{s-1} = \overline{V_0: V_1: \dots V_{s-1}} \end{array} \right) \end{array} \right) \\ \vdots \\ P_{s-1} = \left(\begin{array}{l} R_0 = \dots \left(\begin{array}{l} O_0 = \overline{A_0: A_1: \dots A_{s-1}} \\ \vdots \\ O_1 = \overline{T_0: T_1: \dots T_{s-1}} \\ \vdots \\ O_{s-1} = \overline{F_0: F_1: \dots F_{s-1}} \end{array} \right) \\ \vdots \\ R_1 = \left(\begin{array}{l} M_0 = \overline{I_0: I_1: \dots I_{s-1}} \\ \vdots \\ M_1 = \overline{J_0: J_1: \dots J_{s-1}} \\ \vdots \\ M_{s-1} = \overline{W_0: W_1: \dots W_{s-1}} \end{array} \right) \\ \vdots \\ R_{s-1} = \left(\begin{array}{l} Z_0 = \overline{N_0: N_1: \dots N_{s-1}} \\ \vdots \\ Z_1 = \overline{Y_0: Y_1: \dots Y_{s-1}} \\ \vdots \\ Z_{s-1} = \overline{X_0: X_1: \dots X_{s-1}} \end{array} \right) \end{array} \right) \end{array} \right) \quad (1)
 \end{aligned}$$

де F — інформаційна технологія або система (унітерм F); $P_0, \dots, P_{p-1}$ — підтехнології або підсистеми (унітерми $P_0, \dots, P_{p-1}$) першого рівня декомпозиції унітерму F , p — кількість унітермів першого рівня декомпозиції; $Q_0, Q_1, \dots, Q_{q-1}$ — унітерми другого рівня декомпозиції унітерму F і першого рівня декомпозиції унітерму P_0 , а q — кількість унітермів; $S_0, S_1, \dots, S_{s-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_0, Q_0, \dots$, а s — кількість унітермів; $U_0, U_1, \dots, U_{u-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_0, \dots, S_0$, а u — кількість унітермів; S_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_0, \dots$; $L_0, L_1, \dots, L_{l-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_0, \dots, S_1$, а l — кількість унітермів; S_{s-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_0, \dots$; $K_0, K_1, \dots, K_{k-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_0, \dots, S_{s-1}$, а k — кількість унітермів; $E_0, E_1, \dots, E_{e-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_0, Q_1, \dots$, а e — кількість унітермів; $A_0, A_1, \dots, A_{a-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_1, \dots, E_0$, а a — кількість унітермів; E_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_1, \dots$; $H_0, H_1, \dots, H_{h-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_1, \dots, E_1$, а h — кількість унітермів; E_{e-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_1, \dots$; $B_0, B_1, \dots, B_{b-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_1, \dots, E_{e-1}$, а b — кількість унітермів; $D_0, D_1, \dots, D_{d-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_0, Q_{q-1}, \dots$, а d — кількість унітермів; $C_0, C_1, \dots, C_{c-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_{q-1}, \dots, D_0$, а c — кількість унітермів; D_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_{q-1}, \dots$; $G_0, G_1, \dots, G_{g-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_{q-1}, \dots, D_1$, а g — кількість унітермів; D_{d-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_0, Q_{q-1}, \dots$; $V_0, V_1, \dots, V_{v-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_0, Q_{q-1}, \dots, D_{d-1}$, а v — кількість унітермів; $R_0, R_1, \dots, R_{r-1}$ — унітерми другого рівня декомпозиції унітерму F і першого рівня декомпозиції унітерму P_{p-1} , а r — кількість унітермів; $O_0, O_1, \dots, O_{o-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_{p-1}, R_0, \dots$, а o — кількість унітермів; $A_0, A_1, \dots, A_{\delta-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_0, \dots, O_0$, а δ — кількість унітермів; O_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_0, \dots$; $T_0, T_1, \dots, T_{t-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_0, \dots, O_1$, а t — кількість унітермів; O_{o-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_0, \dots$; $F_0, F_1, \dots, F_{f-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_0, \dots, O_{o-1}$, а f — кількість унітермів; $M_0, M_1, \dots, M_{m-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_{p-1}, R_1, \dots$, а m — кількість унітермів; $I_0, I_1, \dots, I_{i-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_1, \dots, M_0$, а i — кількість унітермів; M_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_1, \dots$; $J_0, J_1, \dots, J_{j-1}$ —

унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_1, \dots, M_1$, а j — кількість унітермів; M_{m-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_1, \dots; W_0: W_1: \dots W_{w-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_1, \dots, M_{m-1}$, а w — кількість унітермів; $Z_0, Z_1, \dots, Z_{z-1}$ — унітерми передостаннього рівня декомпозиції унітерму F і унітермів $P_{p-1}, R_{r-1}, \dots$, а z — кількість унітермів; $N_0: N_1: \dots N_{n-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_{r-1}, \dots, Z_0$, а n — кількість унітермів; Z_1 — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_{r-1}, \dots; Y_0: Y_1: \dots Y_{y-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_{r-1}, \dots, Z_1$, а y — кількість унітермів; Z_{z-1} — унітерм передостаннього рівня декомпозиції унітермів $F, P_{p-1}, R_{r-1}, \dots; X_0: X_1: \dots X_{x-1}$ — унітерми останнього рівня декомпозиції унітермів $F, P_{p-1}, R_{r-1}, \dots, Z_{z-1}$, а x — кількість унітермів; $:$ — розділювач унітермів, який є розділювачем-комою, якщо впорядкованість унітермів немає значення (у цьому випадку операція секвентування є комутативною) або — крапкою з комою, якщо впорядкованість унітермів має значення (операція секвентування є некомутативною).

Рекурентна структурна декомпозиція системи генерування коду.

Формула (2) описує за критерієм функціонального призначення структурну декомпозицію першого рівня інформаційної технології генерування програмного коду з формул алгоритмів. У цій формулі T — унітерм (підсистема) призначений для задання переліку функційних унітермів, які використовуються підсистемами опрацювання операцій алгебри алгоритмів; Se — унітерм опрацювання операції циклічного елімінування, P — унітерм опрацювання операції паралелення, Cr — циклічного паралелення, U — унітермів формул алгоритмів, Cs — циклічного секвентування, E — елімінування, S — секвентування, A — зв'язку з операційною системою, Dcf — задання режимів видалення циклічних операцій, If — задання режимів заміни, Rf — задання режимів підставлення, Zgc — генерування коду, Cef — задання форми циклічного елімінування, Df — задання форми доступу до ключів бази алгоритмів, Mf — головної форми, Sf — задання форми секвентування, Z_s — запису-зчитування, Cpf — задання форми циклічного паралелення, Ef — задання форми елімінування, Mc — оперування з графічними фігурами, Sf — задання форми секвентування, Csf — задання форми циклічного секвентування, Ff — задання шрифтів, Md — задання області для рисування, Sa — задання спеціальних символів унітермів, Def — задання форми видалення циклічних операцій, F — задання параметрів форми, Pf — задання форми операції паралелення, Ug — задання форми унітерму, Df — задання форми доступу до бази алгоритмів, Icf — задання форми заміни складових циклічних операцій, Rf — задання темпів заміни операцій, Wf — видалення графічних фігур, oL — зв'язку підсистеми оптимізації з редактором формул алгоритмів, ooE — оптимізація на основі властивостей елімінування, oC — оптимізація циклічних операцій, oS — оптимізація секвентування, oP — оптимізація паралелення, oEo —, oGi — генерування індексів XML-опису формул алгоритмів, oDd —

задання унітермів-делегатів оптимізації, $oI1$, $oI2$ — задання списків функційних унітермів оптимізації, oBo — базовий унітерм оптимізації, oQ — комунікаційний унітерм оптимізації, oD — оптимізації методом введення додаткової умови.

$$\begin{aligned} & \overbrace{I, C\& P, Cp, U, C\& E, S, A, Dcf, H, Ef, Zec, Ccf, Df, Mf, Sf, Z_{\&}}^{(2)} \\ & \overbrace{Cuf, Et, Me, Sf, Cuf, Ff, Md, Sa, Dcf, F, Pf, L\& Df, Ccf, Ef, Wf} \\ & \overbrace{oI, ooE, oC, oS, oP, oEo, oGi, oDd, oI2, oE, oQ, oD, oI1} \end{aligned}$$

Другого рівня рекурентна декомпозиція унітермів A та Mf виглядає так:

$$A = \overbrace{A_0, A_1, A_2, A_3} \quad Mf = \overbrace{Mf_1, Mf_2, Mf_3, Mf_4}$$

де A_0 та A_1 — підунітерми унітерму A , які наслідують ($:$ — ознака наслідкування) типовий унітерм Ap , що реалізується відомим класом **Application** [3, 6], Mf_0 та Mf_1 — підунітерми унітерму Mf , які наслідують відомий клас **Window** [3, 6].

Третього рівня рекурентна декомпозиція унітермів A_0 , A_1 , Mf_1 :

$$\begin{aligned} & \text{par \% } A_0, Ap = V, N, M, S, T = Mf_1, \quad \text{pu par @ } A_1, Ap = *, \\ Mf_1 = & \left[\begin{aligned} & \overbrace{Z, _CD, bD_C, cD_M, cD_P, cE_C, cP_C, cS_C} \\ & \overbrace{cET_C, cST_C, cZT_C, cT_C, d_C, DS, eIT_C, e_C} \\ & \overbrace{FM_SC, iT_C, MF, MF_P, iT_C, OF, oS_C} \\ & \overbrace{e_C, SF, S_V, S_X, iT_C, e_C, SF, iT, iS, uT, W_A} \\ & \overbrace{uT, uC, iT_C, e_C, iT_C, eIT_C, FM_DC, FM_FC} \\ & \overbrace{pr_C, zJ_C, zS_C, zaT_C, zK_C, zpT_C, RC, R\&} \end{aligned} \right] \end{aligned}$$

де par — ознака частини підсистеми, $\%$ — ідентифікатор опису графіки, pu — ознака загальнодоступності підсистеми, $@$ — ідентифікатор функційної підсистеми, Z — унітерми-змінні, $_CD$ — унітерм вибору унітермів-делегатів, bD_C — перевірки ключів доступу до бази алгоритмів, cD_M — вибору операції алгебри алгоритмів, cD_P — селекції унітермів, cE_C — операції циклічного елімінування, cP_C cP_C — циклічного паралелення, cS_C — циклічного секвентування, cET_C — вибору функційного унітерма циклічного елімінування, cST_C — вибору функційного унітерма циклічного секвентування, cZT_C — вибору функційного унітерма циклічного паралелення, cT_C — вибору шрифту і кегля, d_C — видалення операцій, DS — рисування абстрактного унітерма, eIT_C — вибору функційного унітерма експорту та імпорту до бази алгоритмів, e_C — опрацювання елімінування, eT_C — вибору

функційного унітерма опрацювання елімінування, eI_C — експорту й імпорту до бази алгоритмів, FM_DC — подвійного цоку на головній формі, FM_FC — запису змін, FM_SC — задання початкових значень розмірів форми і сітки, iT_C — інформації про редактор формул алгоритмів, MF — задання початкових значень унітермів-змінних унітерму Mf_1 , MF_P — вибору функційного унітерма рисування, nT_C — перерисування, OF — відкриття файла XML-опису формули алгоритму, oS_C — відкриття файла і задання кегля шрифту, p_C — опрацювання операції паралелення, pr_C — властивості операцій алгебри алгоритмів, RC — перевірки параметрів доступу до бази алгоритмів, Re — рисування початкового абстрактного унітерма, r_C — заміни унітермів, SF — запис формули у вигляді XML-опису, S_V — створення об'єкта відомого класу, S_X — задання смуг прокрутки, sT — вибору функційного унітерма опрацювання секвентування, s_C — функційний унітерм опрацювання секвентування, SF — функційний унітерм видалення графічних об'єктів, tF — обчислення розмірів абстрактного унітерму, tS — зчитування розміру кегля заданого шрифту, uT — вибору функційного унітерму видалення унітермів, W_A — вибору функційного унітерма рисування знаків операцій, wT — вибір функційного унітерма властивостей операцій алгебри алгоритмів, wC — функційний унітерм формування XML-опису доступу до бази алгоритмів, zT_C — закриття форми, zJ_C — вибір функційного унітерма запису файлу із заданням його назви, zS_C — вибір функційного унітерму запису файлу, zaT_C — вибір функційного унітерму заміни, zK_C — вибір функційного унітерма генерування коду, zrT_C — вибір функційного унітерму опрацювання операції паралелення.

Фрагмент рекурентної структурної декомпозиції Mf_0 описується такою формулою

$$Mf_0 = \left(F_0, F_1, F_2, \dots, F_{24} \right),$$

де $F_0, F_1, F_2, \dots, F_{24}$ — параметри унітермів, частково, $F_{22} = Load = «FM_SC»$, у якій є відомою типовою властивістю `Loaded` [3, 6], а FM_SC — назва функційного унітерма опрацювання події `Load`.

Рекурентні структурні декомпозиції унітерму U , Mc та Wf описуються такими формулами:

$$\begin{aligned}
 pu @ U = & \left(\overbrace{Zm, U_0, CS_0, CFC_0, CXML_0, Des_0, Dra_0, GFT_0, GTH_0}^{GTL_0, KF_0, SF_0} \right), \\
 pu @ Mc = & \left(\overbrace{Zc, AV_0, DV_0, GV_0, GVC_0, GV_2_0, GV_C_0, HTC_0}^{HTC_C_0, VCC, N_0} \right),
 \end{aligned}$$

$$Zm @ Wf = SF(), UF(), Wf(),$$

де Zm — унітерми-змінні, $U()$ — функційний унітерм задання початкових значень, $CS()$ — обчислення розмірів унітерма, $CFC()$ — вибору унітерма (селекції), $CXML()$ — створення опису формули алгоритму, $Des()$ — де селекції унітерма, $Dra()$ — рисування унітерму, $GFT()$ — форматування тексту, $GTH()$ — форматування висоти тексту, $GTL()$ — форматування довжини тексту, $KF()$ — обчислення кількості фігур, $SF()$ — видалення фігур, Zc — унітерми-змінні, $AV()$ — доручення об'єкта, $DV()$ — видалення графічних об'єктів, $GV()$ — пошуку потраплянь об'єктів у задану область, $GVC()$ — видача об'єкта за його індексом, $GVs()$ — підготування параметрів і пошук потраплянь, $GV_C()$ — перевірка часткового потрапляння графічних об'єктів у виділену область, $HTC()$ — перевірка ступеня потрапляння об'єкта у виділену область, $HTC_C()$ — перевірка ступеня потрапляння багатьох об'єктів у виділену область, VCC — унітерм — властивість видачі кількості фігур, $N()$ — видачі об'єкта за його номером, $SF()$ — видалення графічних об'єктів, $UF()$ — підготування до видалення графічних об'єктів, $Wf()$ — задання початкових значень, Can — унітермБ, що реалізується відомим графічний елементом Canvas [3; 6], на якому відображаються інші графічні об'єкти.

Функціонування інформаційних технологій і систем починається з вихідної точки. Нею, наприклад, можуть бути ввімкнення системи чи запуск на виконання. У будь-якому випадку має бути виконана умова (u) початку функціонування. Якщо умова не виконана, то відбувається очікування її виконання, що запишемо поверненням у цикл (c_u) за умовою u . Коли ж умова виконана, то функціонування починається з унітерму P . Процес очікування виконання умови опишемо формулою

$$\begin{array}{|c|} \hline u \\ \hline P; c_u; u-? \\ \hline \end{array}.$$

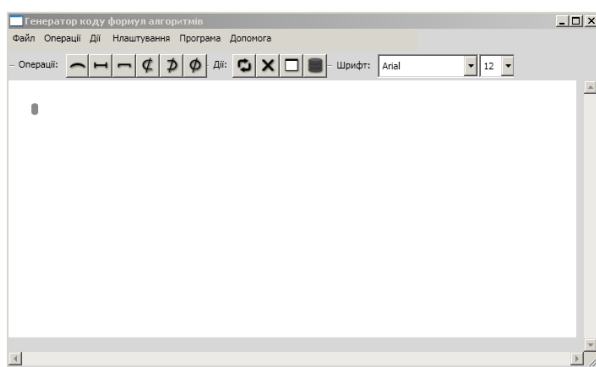
Нехай унітермом P вибирається унітерм A .

Далі розглянемо побудову моделі, у якій усі унітерми ($A, B, C, D, \dots$) мають вибір самих себе (є рекурсивними) і мають вибір кожного з унітермів та кожен з яких може бути вибраним всіма іншими. Опис вибору всіх всіма унітермами складається з двох частин. Перша частина утворена елімінуваннями всіх циклічних секвентувань від унітерму A до всіх решти унітермів ($B, C, D, \dots$). Ці циклічні секвентування запишемо як $u_{ab}, u_{ac}, u_{ad}, \dots$, а їх елімінування запишемо формулою

$$\begin{array}{|c|} \hline u_{ab}; u_{ac}; u_{ad}; \dots; u_{ab}; u_{ac}; u_{ad}; \dots; u_{ab}; u_{ac}; u_{ad}; \dots \\ \hline \end{array}.$$

$$\begin{aligned}
 & \left(\begin{array}{l} Z_{ap}; \\ f_{A1}; \\ Mf_i = \left(\begin{array}{l} Z; \\ MF() = IC() = Mf(); \\ \quad ; @Mc.Zmc \\ \quad RC(); \\ \quad tS_C() = Res(); \\ \quad ; CD_MD() \\ \quad ; \\ \quad piT_C() = tF() = Res() \\ \quad ; \\ \quad Res(); \\ \quad rT = @U.U() = @I.I(); \\ \quad rT.CS() = GIL(); GFT(); GIL(); GFT(); \\ \quad vF = @WF.WF(); \\ \quad vF.UF() = @Mc.GV(); \\ \quad CD_MD() = rT.Dra() = \left(\begin{array}{l} @U.Dra = @Mc.GV(); \\ DRR(); \\ AV() = @Mc.AV(); \end{array} \right. \\ \quad ; \\ \quad CD_MD() = rT.Dra() = \left(\begin{array}{l} @U.Dra() = @Mc.GV(); \\ SF() = \left(\begin{array}{l} @Mc.N(); \\ @Mc.GVC(); \\ @Mc.DV(); \end{array} \right. \\ \quad GV() = Mc.GV(); \\ DRR(); \\ AV() = @Mc.AV() \end{array} \right. \\ \quad ; \\ \quad VCC() = @Mc.VCC() \\ \quad ; \\ \quad GVC() = @Mc.GVC() \end{array} \right. \end{array} \right. \end{aligned} \quad (5)
 \end{aligned}$$

Результатом роботи програмної реалізації рекурсивної функціональної моделі інформаційної технології генерування програмного коду з формул алгоритмів є графічне вікно інтерфейсу користувача, яке показано на рисунку.



Графічне вікно інформаційної технології генерування програмного коду з формул алгоритмів

Отже, створеною рекурентно-декомпозиційною методологією декомпозиційні моделі інформаційних технологій і систем описуються формулами над якими, використовуючи властивості розширеної алгебри алгоритмів, можуть бути виконані тотожні перетворення з метою оптимізації за кількістю унітермів і операцій. Рекурентно-декомпозиційна методологія може застосовуватися для побудови рекурентних декомпозиційних моделей різного виду (структурних, функціональних та інших). Реалізація рекурентних декомпозиційних моделей інформаційної технології генерування програмного коду з формул алгоритмів забезпечує створення графічного інтерфейсу користувача.

1. Великий тлумачний словник сучасної української мови / [уклад. і голов. редактор В. Т. Бусел]. — К.; Ірпінь : Перун, 2002. — 1440 с. 2. Математическая энциклопедия / гл. ред. И. М. Виноградов. — М : Советская Энциклопедия. — 1984. — Т. 4 — 1216 с. 3. Мак-Дональд Мэтью Windows presentation foundation в .NET 3.5 с примерами на C# 2008 / М. Мак-Дональд. — М.; СПб; К.: Apress, 2008. — 922 с. 4. Decomposition (computer science). — Wikipedia, the free encyclopedia [Електронний ресурс]. — Режим доступу: [http://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](http://en.wikipedia.org/wiki/Decomposition_(computer_science)). 5. Owsiak W. Rozszerzenie algebry algorytmów / W. Owsiak, A. Owsiak. // Pomiar, automatyka, kontrola — 2010. № 2. — S. 184–188. 6. Petzold C. Programowanie Windows w języku C# / C. . Petzold. — Warszawa: RM, 2002. — 1161 s.

РЕКУРЕНТНО-ДЕКОМПОЗИЦИОННАЯ МЕТОДОЛОГИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Описана созданная рекуррентно-декомпозиционная методология построения моделей информационных технологий и систем. На ее основе построены структурная и функциональная рекуррентно-декомпозиционные модели информационной технологии генерирования программного кода с формул алгоритмов.

RECURRENT-DECOMPOSITION METHODOLOGY OF INFORMATION TECHNOLOGIES AND SYSTEMS

The created recurrence-decomposition methodology of models of information technology and systems. On its basis created structural and functional recurrent decompositions models of information technology to generate code from formulas of algorithms.

Стаття надійшла 12.07.2011

УДК 621.3

Б. В. Дурняк, І. С. Груник, Л. С. Сікора

Українська академія друкарства

**ОЦІНКА ІНТЕЛЕКТУ ОПЕРАТИВНОГО ПЕРСОНАЛУ
ДЛЯ ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ
В ЕКСТРЕМАЛЬНИХ СИТУАЦІЯХ**

Розглянуто когнітивні аспекти проблеми оцінки інтелекту персоналу, які приймають активні рішення в складних ієрархічних системах.

Інтелект, когнітивна психологія, оперативна діяльність, сенсор.

На сучасному етапі розвитку складних систем з багатомірною розподіленою ієрархічною структурою, насиченою комп'ютерними системами і мережами, для відбору, опрацювання і передачі даних в динамічних потоках між рівнями ієрархії важливим аспектом є оцінка можливостей людини-оператора сприймати інформацію та усвідомлювати її зміст. На основі аналізу проведених досліджень [1-6] можна зробити висновок про те, що ефективне функціонування особи в ієрархічній системі можливе, якщо вона відповідає психічним, когнітивним характеристикам, які є підґрунтям для ефективного навчання здатності прийняття рішень в екстремальних умовах.

Проблема оцінки інтелекту оперативного персоналу для обслуговування складних систем стала особливо важливою на сучасному етапі розвитку комп'ютерних мереж як компоненти складних ієрархічних структур. Помилки інтелектуального характеру (нерозуміння структури і динаміки функціонування телекомунікаційних та комп'ютерних систем) можуть призвести до дезінформації верхнього рівня управління або аварійної ситуації.

Оперативний персонал (ОПР) повинен мати високий рівень мотивації, цілеорієнтації, системного інтелекту для прийняття рішень і їх реалізації в умовах загроз.

Риси особи (ОПР), які характеризують здатність приймати рішення в умовах ризику, пов'язані з певним рівнем інтелекту, який забезпечує можливість і вміння приймати рішення та оцінювати ситуацію:

- читати карти; сприймати образну інформацію;
- прокладати маршрути;
- формувати образи ситуацій;
- розпізнавання ситуацій і осмислення схеми дій;
- генерування тактик дій в умовах ризику;
- сприймати вербальну, образну та аналітичну інформацію, її логічну сутність для класифікації і прийняття рішень.

В умовах ризику в ОПР формується емоційна напруженість. Крім того необхідність швидко і ефективно сприймати і опрацьовувати аналітичну, вимірювальну та образну інформацію, формувати ознаки і їх цілеорієнтовано