

УДК 004.056.55

Л.Е. Серкова, С.Л. Гамоцька

Черкаський державний технологічний університет, Черкаси

ПРОЕКТУВАННЯ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ З ВИКОРИСТАННЯМ АДАПТИВНОГО АНАЛІЗУ

У статті розглянуті основні процедури проектування систем підтримки прийняття рішень з використанням адаптивного аналізу. Запропоновано формальне визначення системи, її компонентів, вхідної і вихідної інформації. Уведено поняття дерева функціональної структури, що використовується для формалізації процедур адаптивного аналізу. Розроблено методи формування дерева функціональної структури, що є науковою та практичною базою для вдосконалення та подальшого підвищення ефективності проектування систем підтримки прийняття рішень.

Ключові слова: система прийняття рішень, система обробки даних, адаптивний аналіз, функціональні підсистеми.

Вступ

При аналізі процесів обробки інформації в існуючих системах підтримки прийняття рішень повинні бути визначені структура системи обробки даних (СОД), вхідні, проміжні і вихідні елементи даних, процедури обробки інформації і послідовність їхньої реалізації, основні характеристики СОД і обмеження на її реалізацію.

При традиційному підході аналіз системи з метою її автоматизації включає наступні етапи:

- виявлення структури системи у вигляді множини функціональних підсистем, причому елементи даної множини можуть утворювати між собою зв'язки "група-підгрупа" і можуть перетинатися;

- виявлення множини функціональних підсистем, що не містить у собі елементів, що включають в себе інші елементи;

- декомпозиція елементів даної множини на більш прості складові доти, поки структура і функції аналізованої системи не стануть очевидними. При цьому глибина структуризації аналізованої системи залежить від можливості синтезувати інформаційну систему у вигляді програмних модулів;

- побудова і структуризація технологічних графів вирішення окремих функціональних блоків обробки даних на основі сукупності графових і матричних моделей, об'єднання окремих графів в узагальнений граф технології.

При використанні процедур адаптивного аналізу здійснюється порівняння необхідних характеристик системи з характеристиками, що зберігаються в бібліотеці прототипів і універсальних програмних модулів. У процесі аналізу можливі два протилежних випадки: у бібліотеці не знайдено жодного функціонального блоку, що підходить для проектованої системи, і у цьому випадку аналіз системи проводиться традиційними методами; у бібліотеці знайде-

ний цілком адекватний прототип, що задовольняє усім вимогам до проектованої системи. У більшості практичних випадків реалізується змішана процедура аналізу, у рамках якої ряд функціональних блоків аналізується шляхом адаптивного порівнювання з інформацією про прототипи бібліотеки, а ряд – традиційними методами.

Постановка завдання. Основними процедурами адаптивного аналізу проектування систем підтримки прийняття рішень є: формування і розмітка дерева функціональної структури проектованої системи; визначення міри близькості бібліотечних програмних модулів до проектованих; прийняття рішення про використання аналогів у системі, що проектується; формування графів технологій обробки даних для унікальних функціональних блоків і їхня адаптація для бібліотечних аналогів; формування узагальненого графа технології обробки даних і його аналіз на повноту реалізації функцій проектованої системи.

Адаптивний аналіз системи, що проектується, ведеться по низхідному принципу, тобто від загального до часткового з послідовним нагромадженням і уточненням інформації про проектовану систему. Так як міра близькості складеного програмного модуля значною мірою залежить від його складових, то її обчислення і формування інтегрованого графа технології обробки даних доцільно здійснювати по висхідному принципу, починаючи з висячих вершин дерева функціональної структури системи.

Основний матеріал досліджень

Першим етапом адаптивного аналізу проектування інформаційної системи є формування дерева ієрархії функціональної структури \hat{F} , вершини якого $\{F\}$ позначені вектором, що містить функціональний опис кожного блоку і його специфікації f від-

повідного рівня агрегації, функціональний блок чи універсальний модуль, що підходить, з бібліотеки m (якщо такі відсутні, то символ пустої множини \emptyset) і міру близькості бібліотечного блоку чи універсального модуля проєктованому $\xi \in [0,1] \cup \lambda'$, де λ' – символ невизначеності, який позначає, що обчислення міри близькості на деякому етапі проєктування не представляється можливим.

Відповідно до низхідного принципу адаптивного аналізу першим його кроком є опис початкової вершини дерева функціональної структури системи F^0 , яка містить найбільш загальні відомості, наприклад, галузь застосування, призначення. На основі даної інформації здійснюється пошук прототипу в бібліотеці L зі співпадаючою чи близькою за значенням вершиною F_1^0 . Якщо аналог не знайдений, то дана вершина дерева позначається вектором $(f^0, 0, 0)$ і здійснюється її декомпозиція на функціональні блоки наступного рівня агрегації F_{k1}^1 , для кожного з яких здійснюється пошук у бібліотеці L , і далі аналогічно. Якщо на деякому етапі підходящий аналог існує в бібліотеці, то він береться в якості "зразка" системи, що проєктується (чи відповідної її частини), тобто аналіз даної функціональної частини системи проводиться по відхиленнях на основі дерева функціональної структури \hat{F}_{pr} з коренем у відповідній вершині бібліотечного функціонального блоку. На даному етапі для усіх вершин дерева \hat{F}_{pr} міра близькості не визначена.

Далі розглядаються усі вершини, суміжні кореневій дерева \hat{F}_{pr} . Для кожної з них перевіряється необхідність реалізації відповідного функціонального блоку в системі, що проєктується. Якщо реалізацій даної функції не потрібно, то відповідна їй вершина і зв'язана з нею гілка видаляються з дерева \hat{F}_{pr} . Далі проводиться аналіз на повноту реалізації функцій системи, що проєктується, на поточному рівні агрегації. Якщо існують функції, що не ввійшли в дерево \hat{F}_{pr} , то вони додаються в якості вершин в структуру, що аналізується, і для них виконується пошук підходящого модуля з бібліотеки. Побудова та обхід дерева ієрархії функціональної структури системи, що проєктується, відбувається на основі алгоритму пошуку по ширині, при якому розгалуження виконується від рівня до рівня. У результаті виконання даних процедур буде сформоване дерево функціональної структури проєктованої системи \hat{F} з вершинами $F_{ik}^j = (f_{ik}^j, m_{ik}^j, \xi_{ik}^j)$. На даному етапі m_{ik}^j і ξ_{ik}^j приймають наступні значення:

$$m_{ik}^j \begin{cases} 0, \text{ коли підходящий аналог у } L \text{ не знайдений,} \\ m \in M \cup \{p_m^{yn}\} \in L, \text{ у іншому випадку;} \end{cases}$$

$$\xi_{ik}^j \begin{cases} 0, \text{ коли } m_{ik}^j = 0, \\ \chi, \text{ у іншому випадку.} \end{cases}$$

Наступним етапом адаптивного аналізу є визначення міри близькості бібліотечного модуля m_{ik}^j , що проєктується для вершин дерева \hat{F} , у яких $m_{ik}^j \neq 0$, і ухвалення рішення про доцільність їхнього використання в системі, що проєктується. Аналіз дерева \hat{F} проводиться по висхідному принципу на основі алгоритму пошуку по ширині, починаючи з нижніх рівнів ієрархії. Міра близькості проєктованого програмного модуля бібліотечному в загальному випадку залежить від міри подібності вхідних і вихідних інформаційних елементів функціонального блоку, набору процедур і складності включення додаткових функцій в складені програмні модулі. Для кожної вершини дерева \hat{F} , що має бібліотечний аналог, обчислимо міру подоби модулів по вхідних інформаційних елементах:

$$\xi_{m_{ik}^j}^{BX} = \frac{2 \left(D_{m_{ik}^j}^{BX} \cap D_{f_{ik}^j}^{BX} \right)}{2 \left(D_{m_{ik}^j}^{BX} \cap D_{f_{ik}^j}^{BX} \right) + \left(D_{m_{ik}^j}^{BX} \setminus D_{f_{ik}^j}^{BX} \right) + \left(D_{f_{ik}^j}^{BX} \setminus D_{m_{ik}^j}^{BX} \right)}.$$

Аналогічним чином обчислюється міра близькості по вихідних інформаційних елементах $\xi_{m_{ik}^j}^{BIX}$ і процедурах $\xi_{m_{ik}^j}^{проц}$. Узагальнена міра подібності проєктованого і бібліотечного простого блоку може бути обчислена за формулою:

$$\xi_{m_{ik}^j} = \frac{P_1 \xi_{m_{ik}^j}^{BX} + P_2 \xi_{m_{ik}^j}^{BIX} + P_3 \xi_{m_{ik}^j}^{проц}}{P_1 + P_2 + P_3}.$$

Значення P_1 , P_2 і P_3 визначають ваги окремих показників у загальній оцінці міри близькості. Для складових програмних модулів (проміжних вершин дерева \hat{F}) узагальнена оцінка буде мати вигляд:

$$\xi_{m_{ik}^j} = \frac{P_1 \xi_{m_{ik}^j}^{BX} + P_2 \xi_{m_{ik}^j}^{BIX} + P_3 \xi_{m_{ik}^j}^{проц} + P_4 \zeta_{m_{ik}^j}}{P_1 + P_2 + P_3 + P_4},$$

де $\zeta_{m_{ik}^j} \notin [0,1]$ – оцінка складності нарощування функціональних можливостей бібліотечного програмного модуля, що визначається експертним шляхом. Дана складова з'являється в узагальненій оцінці у випадку, якщо у вершини $(f_{ik}^j, m_{ik}^j, \xi_{ik}^j)$ є хоча б одна суміжна їй вершина нижнього рівня з $m_{ik}^{j+1} = 0$. Результатом даного етапу адаптивного

аналізу є функціональна структура проєктованої системи \widehat{F} з отриманими значеннями міри близькості проєктованих і бібліотечних програмних модулів ξ_{ik}^j .

На основі даної інформації, задаючи граничне значення міри близькості ξ^* , (як показує практика, $\xi^* = 0,7$), нижче якого адаптація бібліотечних програмних модулів недоцільна (тому, що витрати на адаптацію співвідносні чи більше, ніж розробка унікального модуля), перетворюємо дерево \widehat{F} , замінюючи у вершинах із значенням $\xi_{ik}^j < \xi^*$ імена програмних модулів на 0 ($m_{ik}^j = 0$), а значення міри близькості на 0 ($\xi_{ik}^j = 0$).

Перегляд дерева \widehat{F} на даному етапі здійснюється по спадному принципу на основі алгоритму пошуку по ширині. У результаті виконання даної операції виявляється цілком сформованим дерево функціональної структури системи, що проєктується, визначені функціональні блоки, що мають бібліотечні аналоги, специфікації функціональних блоків, множини вхідних і вихідних інформаційних елементів, множини процедур обробки даних функціональних блоків.

Взаємозв'язок між процедурами обробки даних і інформаційних елементів відбивається за допомогою матриці технології. При аналізі технології обробки даних у СОД і окремих функціональних блоках системи розглядаються не тільки взаємозв'язки між інформаційними елементами, але і використання кожною процедурою обробки даних цих інформаційних елементів. Розглянемо формалізовані методи формування технологічних матриць і графів функціональних блоків СОД.

Нехай $A^F = \{a_r\}$, $r = 1, R^F$ – множина процедур обробки даних функціонального блоку F , $D^F = \{d_l\}$, $l = 1, L^F$ – множина інформаційних елементів F -го блоку. Далі будується матриця технології обробки даних W^F для функціонального блоку F .

У матриці W^F кожен рядок відображає процедуру обробки даних блоку, а кожен стовпець – використання всіма процедурами інформаційного елемента, що розглядається. У рядку міститься інформація про множину вхідних і вихідних даних, зв'язаних з процедурою, що аналізується. Аналіз стовпців дозволяє виявити вхідні і вихідні інформаційні елементи функціонального блоку F . Інформаційний елемент $d_j \in D^F = \{d_l\}$, $l = 1, L^F$, є вхідним, якщо j -й стовпець матриці W^F містить єдиний відмінний від

нуля елемент: $w_{rj} = +1$, $r = 1, R^F$. Якщо j -й стовпець матриці W^F одержить єдиний елемент: $w_{rj} = -1$, $r = 1, R^F$, то відповідний інформаційний елемент $d_j \in D^F = \{d_l\}$, $l = 1, L^F$ є вихідним для блоку F . Ізоморфним матричному представленню взаємозв'язку між процедурами й інформаційними елементами блоку є орієнтований граф технології обробки даних G_T^F .

Узагальненою формою представлення взаємозв'язків процедур і інформаційних елементів при обробці інформації функціональними блоками проєктованої системи є технологічні матриці суміжності і досяжності, що потім перетворюються в інтегрований граф обробки даних.

Далі визначаються вхідні і вихідні елементи для процедур функціонального блоку. Елемент (a_r, d_l) , $r = 1, R^F$, $l = 1, L^F$ матриці W^F дорівнює 1, якщо d_l є вхідним для процедури a_r , і елемент (a_r, d_l) дорівнює -1 , якщо відповідний інформаційний елемент є вихідним. У протилежному випадку елементи матриці B_T , у позиціях (a_r, d_l) і (d_l, a_r) рівні 0. Одиничний елемент у позиції (p_{r1}, p_{r2}) , $r_1, r_2 = 1, R^F$ підматриці B_T^F відповідає наявності одиничних елементів у позиції (p_{r1}, d_l) підматриці B_1^F й у позиції (d_l, p_{r2}) підматриці B_2^F , $l = 1, L^F$, що рівносильно існуванню інформаційного елемента d_l , що є вхідним для процедури P_{r2} і вихідним для процедури P_{r1} .

З використанням матриці суміжності B_T^F , визначається технологічна матриця досяжності M_T^F , що містить підматриці M^F , M_1^F , M_2^F , M_3^F проіндексовані, відповідно множинами (D^F, D^F) , (A^F, D^F) , (D^F, A^F) , (A^F, A^F) .

Підматриця M^F є матрицею досяжності інформаційних елементів функціонального блоку F . Підматриця M_1^F задовольняє співвідношенню:

$$M^F = B_1^F \times M^F = B_1^F \times (B_1^F)^n = B_1^F \times (B^F)^{n+1} \neq B_1^F \times (B^F)^{n-1},$$

де показник ступеня n – ціле додатне число, що не перевищує числа інформаційних елементів функціонального блоку СОД (тобто $n \leq L^F - 1$, і містить одиничні елементи в позиції (p_r, d_l) , якщо процедура P_r , входить у послідовність процедур, необ-

хідних для одержання інформаційного елемента d_i . У протилежному випадку запис у позиції (p_r, d_i) підматриці M_1^F дорівнює 0.

Підматриця M_2^F визначається співвідношенням $M_2^F = M^F \times B_2^F$ і містить одиничний елемент у позиції (d_r, p_1) , якщо інформаційний елемент d_i є входним для послідовності процедур, до складу яких входить процедура P_r . У протилежному випадку елемент (d_r, p_1) підматриці M_2^F , дорівнює 0. Підматриця M_2^F є матрицею досяжності процедур обробки даних і задовольняє співвідношенню:

$$(B_3^F)^{n-1} \neq (B_3^F)^n = (B_3^F)^{n+1} = M_3^F.$$

Одиничний запис у позиції (p_{r1}, p_{r2}) , $r_1, r_2 = 1, R^F$ підматриці M_3^F , відповідає наявності направлено шляху в графі технології обробки даних функціонального блоку F від процедури P_{r1} до процедури P_{r2} .

Побудова єдиного інтегрованого графа здійснюється шляхом виконання операції "накладення" технологічних графів окремих функціональних блоків системи і полягає в сполученні ідентичних рівнів кожного графа й ідентичних вершин на кожному рівні. У результаті формується інтегрований граф, якому відповідає матриця, що отримана шляхом логічного додавання технологічних матриць.

Відповідно до даної технології наступним етапом адаптивного аналізу є формування матриць і графів технологій обробки даних функціональних блоків проектованої СОД і їхнє об'єднання в інтегрований граф обробки даних.

Формування інтегрованого графа проводиться по висхідному принципу з обходом дерева функціональної структури проектованої системи \hat{F} на основі алгоритму пошуку по ширині. На нижньому рівні ієрархії дерева \hat{F} усі вершини є висячими, тобто являють собою прості модулі СОД. При аналізі висячих вершин дерева \hat{F} можливо два випадки: $m_{ik}^j = m \in L$, тобто в бібліотеці є придатний аналог, $m_{ik}^j = 0$ – необхідна розробка унікального модуля.

У першому випадку для формування графа технології обробки даних функціонального блоку F_{ik}^j з бібліотеки L для відповідного програмного модуля m_{ik}^j береться матриця технології обробки даних $W_{ik}^{m_{ik}^j}$, технологічна матриця суміжності $B_T^{m_{ik}^j}$ і модифікуються по наступному алгоритму. Рядки і

стовпці матриць $W_{ik}^{m_{ik}^j}$, $B_T^{m_{ik}^j}$, що відповідають інформаційним елементам, які відсутні у проектованій системі $D_{m_{ik}^j} / D_{f_{ik}^j}$, вилучаються з відповідними їм зв'язками. Аналогічним чином вилучаються рядки і стовпці, що відповідають процедурам, які не використовуються в СОД $A_{m_{ik}^j} / A_{f_{ik}^j}$. Крім того, у мат-

риці $M_T^{m_{ik}^j}$, $B_T^{m_{ik}^j}$ додаються рядки і стовпці, що відповідають інформаційним елементам $D_{m_{ik}^j} / D_{f_{ik}^j}$ і процедурам обробки даних $A_{m_{ik}^j} / A_{f_{ik}^j}$, функціо-

нального блоку F_{ik}^j . Для введених рядків і стовпців проектувальником у діалоговому режимі встановлюються зв'язки попередження/слідування і проводиться перерахунок технологічної матриці досяжності $M_T^{m_{ik}^j}$. Аналіз матриці $M_T^{m_{ik}^j}$ і модифікація матриць $W_{ik}^{m_{ik}^j}$, $B_T^{m_{ik}^j}$ дозволяє в інтерактивному режимі отримати характеристики, що задовольняють вимогам користувачів проектованої системи.

На основі матриці $W_{ik}^{m_{ik}^j}$ формується граф технології обробки даних $G_T^{m_{ik}^j}$ функціонального блоку F_{ik}^j .

При унікальному проектуванні (другий випадок) матриці $W_{ik}^{m_{ik}^j}$, $B_T^{m_{ik}^j}$ і відповідний граф технології обробки даних $G_T^{m_{ik}^j}$ формується стандартними методами.

При аналізі проміжних вершин дерева \hat{F} (складених програмних модулів) формується граф технології обробки відповідного рівня обробки даних аналогічно попередньому випадку. Далі проводиться логічне додавання матриць технології обробки даних $W_{ik}^{m_{ik}^{j-1}}$ і технологічних матриць суміжності $B_T^{m_{ik}^{j-1}}$ рівня $j-1$ для усіх вершин дерева \hat{F} , зв'язаних з вершиною F_{ik}^j .

Отримана матриця $W_{ik}^{m_{ik}^j}$ перетворюється в граф технології обробки даних $G_T^{m_{ik}^j}$. Даний процес продовжується до нульового рівня дерева функціонального опису структури проектованої системи \hat{F} – вершини F_0 .

У результаті виконання процедур даного етапу будуть сформовані інтегрований граф технології

обробки даних СОД G_T^0 , відповідні матриці технології обробки даних W^0 і технологічні матриці суміжності V_T^0 і досяжності M_T^0 .

Заключним етапом аналізу і проектування систем є перевірка коректності розробленої структури системи обробки даних, представленої у виді інтегрованого графа технології обробки G_T^0 і сукупності матричних моделей.

Граф G_T^0 не повинен мати ізольованих вершин, тобто всі інформаційні елементи системи $d \in D_p$ повинні бути вхідними чи вихідними для деякої процедури обробки даних, а будь-яка процедура обробки даних $a \in A_p$ повинна мати хоча б один вхідний і один вихідний інформаційний елемент СОД, тому що за визначенням процедури обробки даних являють собою відношення між інформаційними елементами. Аналіз технологічної матриці суміжності V_T^0 дозволяє виділити незв'язані (надлишкові) елементи системи обробки даних. Елемент СОД $u \in D_p \cup A$ надлишковий у структурі системи тоді і тільки тоді, коли рядок і стовпець і матриці V_T^0 містить тільки нульові елементи, за винятком елемента (u, u) . Усі дані елементи не використовуються, тому можуть бути вилучені зі структури СОД викреслюванням даних рядків і стовпців у матриці V_T^0 і подальшою реструктуризацією інтегрованого графа технології G_T^0 .

Розроблена структура системи обробки даних повинна задовольняти інформаційним вимогам

користувачів системи, тобто забезпечувати формування заданих ними інформаційних елементів (груп інформаційних елементів) по визначених алгоритмах. Дана задача може бути вирішена на основі аналізу технологічної матриці досяжності.

Висновки

Запропоновано формальне визначення системи, її компонентів, вхідної і вихідної інформації. Уведено поняття дерева функціональної структури, що використовується для формалізації процедур адаптивного аналізу.

Розроблено методи формування дерева функціональної структури, що є науковою та практичною базою для вдосконалення та подальшого підвищення ефективності проектування систем підтримки прийняття рішень.

Список літератури

1. Петров Е.Г. Методы и средства принятия решений в социально-экономических и технических системах / Е.Г. Петров. – Херсон: ОЛДИ-Плюс, 2003. – 320 с.
2. Кобелев П.Б. Практика применения экономико-математических методов и моделей / П.Б. Кобелев. – М.: Финстатинформ, 2002. – 246 с.
3. Норенков И.П. Основы автоматизированного проектирования / И.П. Норенков. – М.: МГТУ им. Н.Э. Баумана, 2002. – 336 с.

Надійшла до редколегії 12.09.2011

Рецензент: д-р техн. наук проф. І.В. Шостак, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.

ПРОЕКТИРОВАНИЕ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ АДАПТИВНОГО АНАЛИЗА

Л.Е. Серкова, С.Л. Гамоцкая

В статье рассмотрены основные процедуры проектирования систем поддержки принятия решений с использованием адаптивного анализа. Предложено формальное определение системы, ее компонентов, входной и выходной информации. Введено понятие дерева функциональной структуры, которая используется для формализации процедур адаптивного анализа. Разработаны методы формирования дерева функциональной структуры, которая является научной и практической базой для совершенствования и последующего повышения эффективности проектирования систем поддержки принятия решений.

Ключевые слова: система принятия решений, система обработки данных, адаптивный анализ, функциональные подсистемы.

PLANNING THE DECISION SUPPORT SYSTEMS WITH USING OF AN ADAPTIVE ANALYSIS

L.E. Serkova, S.L. Gamotska

In the article considered basic procedures of planning the decision support systems with using of an adaptive analysis. Proposed a formal determination of the system, its components, input and output information. Introduced the concept of the tree of functional structure which is used to formalize the procedures of an adaptive analysis. Developed the methods for the formation of the tree of functional structure which is a scientific and practical base for improving and subsequent increase of efficiency of planning the decision support systems.

Keywords: system of decision-solutions, data processing system, adaptive analysis, functional subsystems.