ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.422.8

д.т.н., доц. **Бойченко О.В.** (ОДАТРЯ)
д.т.н., проф. **Лєнков С.В.** (ВІКНУ)
Охрамович Л.В. (ВІКНУ)

МОДУЛЬНЕ ПРОЕКТУВАННЯ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті запропоновано застосування модульного принципу для створення можливості підвищення стійкості прикладного програмного забезпечення інформаційної системи за рахунок виникнення численних природних контрольних точок для спостереження за просуванням програми. Основою модульного проектування є функціональна декомпозиція, коли шари проектованої системи стають рівнями модулів у програмі, відбиваючи майбутню ієрархію системи. На етапі модульного проектування структурно алгоритмічним підходом здійснюється пошарове проектування прикладного програмного забезпечення інформаційної системи, що за рахунок декомпозиції дозволяє створити умови для максимально ефективного контролю усієї модульно-логічної структури системи. Зазначене визначає шлях побудови методологічних основ для контролю несанкціонованих впливів на прикладне програмне забезпечення інформаційної системи.

Ключові слова: модульне проектування, стійка інформаційна система, прикладне програмне забезпечення, несанкціоновані впливи.

Постановка проблеми. Одним з основних завдань функціонування інформаційних систем (ІС) спеціального призначення є забезпечення безперебійного отримання достовірної, актуальної та коректної інформації для підтримки управлінських рішень. Зазначене визначає основну вимогу до програмного забезпечення ІС – коректність, яка досягається з точки зору її стійкості.

Сучасні методології структурного проектування прикладного програмного забезпечення (ППЗ), які орієнтовані на оброблення, надають особливого значення процесу і структурі в створенні архітектури програми.

Аналіз останніх досліджень та публікацій. Роботи відомих науковців у сфері дослідження та розроблення відмово стійких та стійких інформаційних систем реального часу, що функціонують в умовах невизначеності та ризику даних вказують на доволі великий перелік методів та засобів підвищення живучості інформаційної системи (Сбітнєв А.І., Барабаш О.В. та ін.). С точки зору структуризації проектування елементів ППЗ, найбільш прийнятними є концепції модульного програмування [59], перевагами яких є відомі позитивні якості модульних програм:

легкість упорядкування і налагодження. Функціональні компоненти такої програми пишуться й налагоджуються порізно. Можливе добре структуроване налагодження як «нагору», так і «униз»;

легкість супроводу і модифікації;

можливість розподілу модулів між програмістами різноманітної кваліфікації відповідно до рівня складності;

можливість створювати бібліотеки найбільше вживаних програм;

спрощення процедури завантаження в оперативну пам’ять великої задачі, що потребує сегментації;

виникнення численних природних контрольних точок для спостереження за просуванням програми, використовуваних для підвищення усталеності ПЗ.

Постановка завдання. Метою роботи є розроблення ППЗ ІС за модульним принципом для створення можливості підвищення стійкості ППЗ за рахунок виникнення численних природних контрольних точок для спостереження за просуванням програми.

Виклад основного матеріалу. Передача керування в модульній структурі відбуваються лише по вертикальних лініях, що з’єднує модулі в схемі ієрархії.

Будь-який модуль може активізувати підпорядкований модуль і одержати керування після завершення його роботи. Модуль більш низького прошарку не може викликати модуль більш високого прошарку.

Даний етап по суті є етапом функціональної декомпозиції. Шари системи, що проектується, на цьому етапі стають рівнями модулів у програмі, що створюється, відбиваючи майбутню ієрархію системи.

До цього часу існують різночитання поняття модульності. У роботі прийнято визначення, яке наведене у [91], згідно з яким модульним називається таке програмне забезпечення, в якому можна змінити будь-яку частину логічної структури, не викликаючи змін в інших частинах.

Під модулем розуміється послідовність логічно зв'язаних фрагментів, які оформлені як окрема частина програми.

Визначальна властивість модуля – незалежність щодо логічної структури прикладного програмного забезпечення (ППЗ), аргументів (параметрів) модуля, констант тощо.

Розрізняють Ф-модулі (акції - функціональні дії), І-модулі (інтерфейси) і ОСД-модулі (операції над структурами даних).

Введення ОСД-модулів пояснюється тим, що правильне використання структур даних грає дуже важливу роль при проектуванні, вони визначають його ефективність.

На конструкцію модуля накладаються такі обмеження:

модуль повинен мати один вхід і один вихід;

модуль не повинен змінювати команди іншого модуля;

модуль не повинен зберігати історію своїх викликів (хоча її наявність спрощує трасування при збоях роботи ППЗ);

в ідеалі кожний модуль повинен реалізувати одну функцію цілком.

Суттєві обмеження на глибину модульності програм накладає підвищення витрат обчислювальних ресурсів системи за часом роботи центрального процесора й обсягом пам'яті.

Додаткові обмеження на реалізацію модульних програм полягають в наступному.

Передачі управління між модулями відбуваються по вертикальних лініях, що з'єднують модулі у схемі ієрархії. Модуль, що активізує підлеглий модуль, отримує управління після завершення його роботи (зворотне не вірно – модуль більш низького шару не може викликати модуль більш високого шару). Модуль завжди повертає управління тому модулю, що його викликав.

Прийняття рішень модулями нижчого рівня за модулі вищого рівня не допускається. Надто велика кількість аргументів, що надаються модулю, тягне необхідність розподілу функцій.

Пропонується навіть обчислювати кількість модулів, виходячи з залежності $M = \eta / 6$, де M – кількість модулів, η – кількість єдиних за змістом вхідних і вихідних параметрів, які включені в алгоритм. Дана залежність базується на припущенні, що кількість параметрів ідеального модуля дорівнює 6.

Конструкція модулів складається з двох частин: оголошення даних і оголошення акцій. Розрізняють дані трьох типів: глобальні, локальні і формальні параметри акцій. Глобальні дані реалізують зв'язок задачі із зовнішнім середовищем (їх початкові значення встановлюються перед виконанням програми). Локальні дані оголошуються всередині акції й активізуються при її роботі. Акції модуля реалізують дії, що задаються програмістом, і можуть мати формальні параметри.

Введемо такі позначки для відношень між модулями: r_1 - «x викликає y», r_2 - «x здійснює доступ до y», r_3 - «x читає y», r_4 - «x захищений від y», r_5 - «x уточнює y». Домовимося також виникнення відношень між модулями, що ініціюються ситуацією S,

позначати через $r_k(S)$. Основні відношення між модулями та їх інтерпретація зведені в табл. 1.

Таблиця 1

Основні відношення між модулями

Позначення	Інтерпретація
$M_k \; r_1 \; M_1$	Акції модуля M_k містять виклики акцій з модуля M_1 .
$M_k \; r_2 \; M_1$	Акції M_k використовують глобальні дані M_1 й (можливо) змінюють їх.
$M_k \; r_3 \; M_1$	Акції M_k читають глобальні дані модуля M_1 без їх змін.
$M_k \; r_4 \; M_1$	Дії усередині M_1 не можуть змінювати дані, що зберігаються в M_k.
$M_k \; r_5 \; M_1$	M_1 – модуль структури даних, M_k – діє за допомогою структурної операції.
$M_k \; r_1(S) \; M_1$	Акції модуля M_k активізуються в ситуації S.
$r_2(S) \; M_k$	В ситуації S змінюються глобальні дані модуля M_k.

Тоді вимоги, що ставляться до модульної структури в рамках методології проектування, можуть бути сформульовані так.

А. Акції модуля можуть завжди використовувати дані того модуля, частинами якого вони є:

$$M_k \; r_2 \; M_k .$$

Б. Доступний модуль повинен бути таким, що читається:

$$M_k \; r_2 \; M_1 \leftrightarrow M_k \; r_3 \; M_1 .$$

В. Акції верхніх модулів можуть викликати акції нижніх модулів, зворотне заборонено:

$$M_k \; r_1 \; M_1 \rightarrow (M_k \neq M_1) \,\&\, (M_1 \; r_1 \; M_k).$$

Г. Має місце ієрархічне упорядкування:

$$\text{П}\, r_1 \,\text{З}\; r_1 \; \text{ФЯ} \; r_1 \; M_1^{\,u} \; r_1 \; \ldots \; r_1 \; M_i^{\,u} \; r_1 \ldots r_1 \; M_{ф} \;,$$

де П - процес, З - задача, ФЯ - функціональне ядро, $M_i^{\,u}$ - *І*-модулі, $M_{ф}$ - Ф-модуль.

Д. Верхні модулі звертаються до даних нижніх модулів шляхом виклику акцій, що мають доступ до цих даних:

$$M_k \; r_1 \; M_l \; \rightarrow \; M_k \urcorner r_3 \; M_l \;,$$

а винятком, коли

$$\text{З}_p \; r_1 \; \text{З}_q \;\; (p \neq q) \rightarrow \text{З}_p \; r_3 \; \text{З}_q .$$

Е. Задачі, що не є процесами, не запускаються при настанні зовнішніх подій (виникненні ситуацій):

$$\text{З}_p \; r_1 \; \text{З}_q \rightarrow \urcorner r_1 \; (S) \; \text{З}_q .$$

Ж. Існує тип відношень між елементами, що припускають порушення принципів модульного проектування:

$$\urcorner r_1(S) \text{П}\, r_1 \; \text{З}_{M_k} \; r_1 \; \text{З}_{M_l} \rightarrow r_1(S) \text{П}\, r_1(S) \; \text{З}_{M_k} \; r_2(S) \text{З}_{M_l} \; r_2(S) \text{ФЯ} \; r_2(S) \; r_2(S) \ldots$$

$$r_2(S) \, M_{осд} .$$

Його реалізація диктується необхідністю забезпечення високої реактивності системи у ситуаціях, що виникають. При цьому доступ до даних здійснюється найчастіше через загальну область пам'яті.

Таким чином основними моделями, що їх використовує технологія структурного проектування, є графові моделі.

Отримані наукові результати. Наведені співвідношення дозволяють формалізувати процес верифікації модульної структури ППЗ, що розробляється.

Введення певних правил модульного проектування, опис їх у термінах формальної теорії дозволяє побудувати формальні моделі ієрархічних модульних структур і провести їх верифікацію.

Висновки. На етапі модульного проектування структурно алгоритмічним підходом здійснюється пошарове проектування ППЗ ІС, що за рахунок декомпозиції дозволяє створити умови для максимально ефективного контролю усієї модульно-логічної структури системи. Зазначене визначає шлях побудови методологічних основ для контролю несанкціонованих впливів на ППЗ ІС.

ЛІТЕРАТУРА:

1. Сбітнєв А.І. Структурне проектування спеціального програмного забезпечення / А.І. Сбітнєв, С.В. Ленков, О.М. Грищак // Вісник Східноукраїнського національного університету ім. В. Даля. – Луганськ: СНУ, 2007. – Ч. 1. – № 5 (111). – С. 147-154.

2. Сбитнев А.И. Структурные методы проектирования математического обеспечения АСУ ТП / А.И. Сбитнев // Модели и алгоритмы автоматизированных систем в промышленности. – К.: ИК АН УССР, 1982. – С. 3-9.

3. Пайк Х. Разработка программного обеспечения для мини-ЭВМ / Х. Пайк // Мини-ЭВМ. – М.: Мир, 1975. – С. 27-41.

4. Бойченко О.В. Структурне проектування програмного забезпечення складних інформаційних систем реального часу / О.В. Бойченко С.В. Лєнков, П.А. Шкуліпа // Сучасна спеціальна техніка. – К.: ДНДІ МВС України, 2012. – № 4(31). – С. 92-97.

5. Лингер Р. Теория и практика структурного программирования / Р. Лингер, Х. Миллс, Б. Уитт; [пер. с англ.]. – М.: Мир, 1982. – 406 с.

6. Сбитнев А.И. Декомпозиция программного обеспечения в многомашинных системах управления: Системный анализ / А. И. Сбитнев, Б.Д. Шепетюк. – К.: ИК АН УССР, 1982. С. 26–52.

Без рецензії.

д.т.н., доц. Бойченко О.В., д.т.н., проф., Ленков С.В., Охрамович Л.В.

МОДУЛЬНОЕ ПРОЕКТИРОВАНИЕ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье предложено применение модульного принципа для создания возможности повышения стойкости прикладного программного обеспечения информационной системы за счет возникновения многочисленных естественных контрольных точек для наблюдения за продвижением программы. Основой модульного проектирования является функциональная декомпозиция, когда слои проектируемой системы становятся уровнями модулей в программе, отбивая будущую иерархию системы. На этапе модульного проектирования структурно алгоритмическим подходом осуществляется послойное проектирование прикладного программного обеспечения информационной системы, что за счет декомпозиции позволяет создать условия для максимально эффективного контроля всей модульно-логической структуры системы. Полученные результаты определяют путь построения методологических основ для контроля несанкционированных влияний на прикладное программное обеспечение информационной системы.

Ключевые слова: модульное проектирование, стойкая информационная система, прикладное программное обеспечение, несанкционированные влияния.

Boychenko O., Lenkov S., Ohramovich L.

MODULE PLANNING OF APPLICATION SOFTWARE

In the article module principle is offered for creation of possibility of increase of firmness of application software of the informative system due to the origin of numerous natural control points for looking after advancement of the program.

Basis of the module planning is a functional decoupling, when the layers of the designed system become the levels of the modules in the program, removing the future hierarchy of the system.

On the stage of the module planning structurally algorithmic approach is carry out the layer planning of application software of the informative system, that due to a decoupling allows to create terms for maximally effective control all module-logical structures of the system. The marked determines the way of construction of methodological bases for control of unauthorized influences on application software of the informative system.

Keywords: module planning, proof informative system, application software, unauthorized influences.