



## СТРУКТУРНИЙ СИНТЕЗ ДИНАМІЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Світлана Антощук, Дмитро Маєвський

Одеський національний політехнічний університет  
просп. Шевченка, 1, 65044, м. Одеса,  
e-mail: asg@ics.opu.ua, Dmitry.A.Maevsky@gmail.com

**Резюме:** Розглянуто теоретичні основи та створено метод синтезу структури динамічних інформаційних систем із здатністю до адаптування при змінах предметної області. Здатність до адаптування забезпечується за рахунок виділення варіативної частини алгоритмів системи, що змінюються при змінах предметної області. Варіативна частина алгоритмів зберігається в базі даних системи, що забезпечує їх швидку заміну та підвищує надійність системи на етапі експлуатації.

**Ключові слова:** динамічна інформаційна система, синтез структури, варіативні алгоритми, надійність.

## STRUCTURAL SYNTHESIS OF DYNAMIC INFORMATION SYSTEMS

Svetlana Antoschuk, Dmitry Maevsky

Odessa National Polytechnic University  
1, Shevchenko boulevard, 65044, Odessa, Ukraine  
e-mail: asg@ics.opu.ua, Dmitry.A.Maevsky@gmail.com

**Abstract:** Theoretical bases are considered and the method of synthesis of the dynamic information systems structure is created with a capacity for adaptation at the changes of a subject domain. A capacity for adaptation is provided due to the selection of a variation part of algorithms of the system that changing at the changing the subject domain. The variation part of algorithms is kept in the database of the system that provides their rapid replacement and promotes a fail safety on the stage of exploitation.

**Keywords:** dynamic information system, synthesis of structure, variant algorithms, reliability.

### ВСТУП

Інформаційні системи, предметна область (ПрО) яких зазнає змін на етапі експлуатації є дуже розповсюдженими. Будемо називати такі системи динамічними інформаційними системами (ДІС). Типовими представниками ДІС є облікові інформаційні системи, ПрО яких є господарче законодавство України. За даними Державного управління статистики, загальна кількість підприємств, що використовують облікові системи сягає більш ніж дев'ятисот тисяч. Водночас, як показали дослідження, закони та підзаконні акти в цій сфері змінюються в середньому один раз на десять днів [1]. Така частота змін, зважаючи на дуже велику кількість ДІС, породжує проблему забезпечення їх безперервної роботи під час експлуатації.

Простої ДІС, що виникають при їх оновленні, негативно впливають на їх надійність, можуть призвести до призупинення виробничих процесів та значних матеріальних втрат. Збереження надійності на цьому етапі можливо лише за рахунок зниження часу простоїв при оновленнях.

Сучасні підходи до структурної побудови усіх ІС передбачають інтеграцію усіх програмних модулів у один великий за обсягом файл [2, 3]. Таку, побудовану за традиційним підходом структуру ІС назвемо статичною структурою.

Треба відмітити, що саме статична структура побудови сприяє виникненню негативного фактору, що збільшує час простоїв ДІС при оновленнях. Дійсно, усі алгоритми, що реалізують функціональні можливості ДІС, входять до складу їх ПЗ та не невід'ємними від нього. При оновленнях, що торкаються, зазвичай

не усієї множини алгоритмів, тобто частки ПЗ, зміни потребує усе воно разом. Це призводить до виникнення таких негативних факторів. По-перше, обсяг інформаційного пакету, що потрібен для оновлення такої ДІС не виправдано збільшується. По-друге, збільшується, відповідно, й час, що потрібен для передачі пакету оновлень до користувачів. У період призупинення для оновлення ДІС частково або повністю перестає виконувати свої функції, що також сприяє зменшенню їх надійності та виникненню ризиків аварій та техногенних катастроф, особливо у галузях безперервного циклу роботи (енергетика, транспорт, банківські системи). По-третє, передача великого обсягу інформації до великої кількості користувачів обмежена, пропускну здатністю існуючих каналів зв'язку. Для з'ясування істотності проблем, що породжуються цим фактором, розглянемо ОІС «Бухгалтерський облік для України», яку розроблено фірмою «ІС». Усі програмні модулі та екранні форми цієї ОІС розміщено в одному компаунд-файлі розміром близько десяти мегабайт [4]. Ця ОІС широко використовується більшістю підприємств та організацій України (більше, ніж дев'ятисот тисяч). Тобто, при оновленнях цієї ОІС до серверу з файлом оновлень розміром у десять мегабайт намагатимуться отримати доступ кілька сотень тисяч абонентів одночасно. Зважаючи на обмежену пропускну здатність каналів зв'язку, виникає проблема зниження надійності ДІС за рахунок збільшення часу на їх обслуговування ДІС.

Необхідність передачі значного за обсягом пакету оновлень до великої кількості користувачів фактично еквівалентна розподіленій атаці типу «відмова в обслуговуванні» (DDoS – атаці) на сервер провайдера, де розміщені оновлення. Як відомо, атаки такого типу можуть призвести повної відмови в обслуговуванні усіх клієнтів, які користуються послугами цього провайдера, що призводить до дуже значних фінансових втрат. Так, за дослідженнями компанії Forrester Research, проведеними у 2009 році, якщо на сервері розміщені WEB-сервіси електронної комерції, то призупинення роботи такого сервера коштує від 190 до 650 тисяч доларів за годину.

Таким чином, на етапі експлуатації ДІС виникають такі протиріччя:

– протиріччя між потребою зменшення простоїв ДІС на етапі експлуатації та необхідністю тривалого призупинення їх роботи для виконання оновлень при змінах ПрО;

– протиріччя між необхідністю одночасної передачі великих за обсягом інформаційних

пакетів для оновлення ДІС до великої кількості користувачів, та обмеженою пропускну здатністю каналів зв'язку, що призводить до зниження надійності ДІС на етапі експлуатації за рахунок збільшення часу простоїв при оновленнях.

Розв'язати ці протиріччя можна за рахунок спеціально побудованої структури ДІС, яка дозволяє адаптування системи до змінних вимог предметної області. Тому задача теоретичного обґрунтування та створення методів побудови такої структури ДІС є актуальною.

## 1. ОСНОВНА КОНЦЕПЦІЯ СТРУКТУРНОЇ ПОБУДОВИ ДІС

Основна концепція побудови структури ДІС, яка дозволяє швидку адаптацію при зміні вимог ПрО, полягає в виділенні із всієї множини алгоритмів обробки інформації в ДІС її варіативної підмножини, саме яка й зазнає змін при змінах ПрО. Основною особливістю запропонованої структури є зберігання варіативної підмножини алгоритмів не в загальному програмному коді, а відокремлено від нього, в інформаційній базі системи у вигляді не трансльованого або частково трансльованого коду (так званого «р-коду»). Такий код займає значно менше місця, ніж відповідний йому трансльований двійковий код, що вирішує проблему його швидкої передачі до великої кількості користувачів, що, як показали дослідження, до двох разів зменшує коефіцієнт простоїв ДІС при оновленні.

На самому вищому рівні будь-яку ІС можна розглядати як сукупність двох складових: програмного забезпечення та інформаційної бази. Кожна з цих складових не існує сама по собі. Для їх об'єднання в систему між ними повинні існувати певні взаємозв'язки – певні частини ПЗ певним образом співвідносяться із певними об'єктами ІБ. Тоді модель структури будь-якої ІС можна представити як трійку:

$$S = \{A, I, L\}, \quad (1)$$

де  $A$  – множина усіх алгоритмів ДІС,  $I$  – множина усіх структурних одиниць інформаційної бази (ІБ), а  $L$  – множина усіх зв'язків між алгоритмами та структурними одиницями ІБ.

Але у випадку ДІС, при змінах ПрО, зміни торкаються тільки певної підмножини  $A_v$  варіативних алгоритмів, що змінюються:  $A_v \in A$ . Проведений аналіз показав, що потужність множини  $A_v$  складає близько п'яти відсотків від потужності  $A$ . Так, наприклад, в обліковій ДІС

«АгроКомплекс» [5], міститься загалом 7513 програмних модулів, що реалізують алгоритми ДІС. З них варіативна частина складає 360 алгоритмів, тобто лише 4,8 відсотки.

Різницю

$$A_c = A \setminus A_v, \{a_c \in A \mid a_c \notin A_v\}$$

будемо називати множиною сталих алгоритмів, що не змінюються при змінах ПрО.

У свою чергу, множина  $I$  для ДІС може бути представлена як  $I = T \cup R$ , де  $T$  – множина усіх інформативних таблиць ІБ, а  $R$  – множина накопичувальних реєстрів. Тому модель (1) можна представити як:

$$S = \{\{A_c, A_v\}, \{T, R\}, L\}. \quad (2)$$

Згідно з цією моделлю, при змінах ПрО ДІС, зміні підлягають усі алгоритми множини  $A = A_c \cup A_v$ , у той час, як реально змінюється не більше п'яти відсотків алгоритмів. Тому включення варіативної підмножини  $A_v$  до  $A$  призводить до протиріччя між необхідністю одночасної передачі великих обсягів інформації для оновлення ДІС до великої кількості користувачів та обмеженою пропускною здатністю каналів зв'язку. Це, у свою чергу, викликає зниження надійності ДІС за рахунок збільшення часу їх простоїв під час оновлення.

Для розв'язання цього протиріччя зробимо перегруповування множин  $A$  та  $I$  таким чином:

$$A' = A \setminus A_v.$$

Тут нова множина  $A'$  утворена як різниця між загальною множиною усіх алгоритмів ДІС та множиною варіативних алгоритмів. Фактично, цією операцією ми відокремлюємо варіативну частину від множини усіх алгоритмів, отримуючи окрему сталу їх частину:

$$A' = A_c.$$

Залишок від різниці  $A \setminus A'$  являє собою варіативну частину алгоритмів, яку включимо до множини  $I$ , утворюючи нову розширену множину  $I'$ :

$$I' = I \cup (A \setminus A').$$

Тоді структуру (2) можна переписати як:

$$S = \{\{A_c\}, \{T, R, A_v\}, L\}. \quad (3)$$

Структура (3) відрізняється від традиційної структури (2) тим, що варіативні алгоритми переносяться до складу інформаційної бази ДІС. Відокремлення множини сталих алгоритмів за моделлю (3) дозволяє звести процес оновлення ДІС лише до оновлення тієї варіативної підмножини  $A_v$ , що змінилася. При чому відмітимо, що тепер при оновленнях зміні підлягають навіть не усі варіативні алгоритми, а

саме та їх частка, що змінилася. Таке оновлення потребує передачі по каналах зв'язку значно меншого обсягу даних (декілька десятків кілобайт), що призводить до значного зменшення часу простоїв ДІС при оновленнях.

Розглянемо гнучку структуру ДІС (рис. 1), що відповідає моделі (3) та дозволяє виконувати адаптацію до змін ПрО [6].

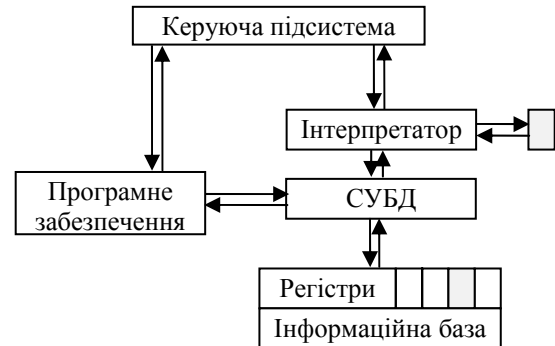


Рис. 1 – Структура ДІС з відокремленим програмним кодом

Робота ДІС із змінною структурою відбувається таким чином [7, 8]. Система отримує запит користувача на виконання дії, що потребує роботи певного алгоритму. Цей запит, як і в попередньому випадку традиційної побудови ДІС, аналізується керуючою підсистемою, яка визначає, до якої підмножини алгоритмів відноситься цей алгоритм – до варіативної або сталої.

У випадку, коли алгоритм відноситься до варіативної множини, керуюча підсистема видає відповідний запит до СУБД, яка знаходить в інформаційній базі потрібний програмний код варіативного алгоритму.

Але далі процес змінюється. До складу ДІС, яку побудовано за новою структурою із можливістю адаптування до змін ПрО входить нова складова – інтерпретуюча підсистема. Знайдений в ІБ ДІС код варіативного алгоритму передається на виконання до цієї інтерпретуючої підсистеми. Як результат його виконання може змінюватись стан інформаційної бази ДІС або формуватись звіти для користувача.

У випадку, коли дія системи повинна виконуватись статичним алгоритмом, робота ДІС відбувається так же, як і завжди. В цьому випадку система просто звертається до потрібного програмного модуля, що входить до складу ПЗ ДІС, а результат його виконання змінює (при потребі) стан ІБ, або видається користувачеві.

Необхідність застосування інтерпретатора та пов'язане з цим можливе зниження швидкодії системи не може вважатись вадою

запропонованого підходу. Проведені численні дослідження довели, що зниження швидкодії за рахунок інтерпретації не є суттєвим [9, 10, 11].

Керуюча підсистема отримує запит користувача на виконання певної дії та визначає, до якої множини алгоритмів належить алгоритм, який повинен виконувати цю дію. Якщо обрано алгоритм із множини варіативних, керуюча підсистема видає відповідну інформацію до СУБД, яка знаходить в інформаційній базі потрібний програмний код та передає його на виконання до інтерпретуючої підсистеми. На рис. 1 програмні модулі варіативної підмножини показано сірим кольором. Інтерпретатор виконує цей програмний код. Як результат його виконання може змінюватись стан інформаційної бази ДІС або формуватись звіти для користувача. У випадку вибору алгоритму із множини статичних, робота ДІС виконується традиційним чином – керування передається відповідній ділянці програмного коду, що є складовою ПЗ ДІС.

## 2. МЕТОД СТРУКТУРНОГО СИНТЕЗУ ДІС

Для побудови гнучкої структури ДІС із здатністю до адаптації розроблено відповідний метод, який полягає в виділенні варіативної частини алгоритмів на підставі аналізу ПрО, та реалізується послідовністю взаємопов'язаних етапів, кожен з котрих розписано на рівні операцій та елементарних дій (рис. 2).

Метод полягає в послідовному виконанні таких етапів: структуризація системи; моделювання управління підсистемами; декомпозиція підсистем на структурні одиниці; виділення структурних одиниць, що змінюються; розподіл структурних одиниць.

Етап 1 починається із аналізу технічного завдання на розробку ДІС та її ПрО. Визначається множина елементів ПрО та провадиться вибір тієї підмножини, що відображається в ДІС, що проектується. Необхідність цього етапу пов'язана з тим, що ПрО ДІС є більш ширшою, аніж ДІС, що відображає цю ПрО. ДІС здатна відобразити лише формалізовані взаємозв'язки між об'єктами її ПрО.

Після вибору підмножини об'єктів ПрО, що буде відображатися у ДІС, провадиться їх систематизація (кластеризація) за ідентифікаційною ознакою кластерів. Наприклад, в облікових ДІС такими кластерами об'єктів є кластери «Бухгалтерський облік», «Податковий облік», «Оперативний облік». Це виділення кластерів й є змістом операції 1.1 на рис. 2. На

підставі аналізу ПрО та вимог ТЗ, обрані на цьому етапі кластери, й будуть визначати структуру майбутньої ДІС.

В операції 1.2 виконується виявлення взаємозв'язків між обраними на попередньому етапі кластерами та формування підсистем майбутньої ДІС. В нашому випадку такими підсистемами виступатимуть обрані кластери ПрО – «Бухгалтерський облік», «Податковий облік», «Оперативний облік».

На цьому етапі слід мати на увазі, що кластери ПрО та відповідні їм підсистеми ДІС не існують самі по собі, а мають між собою тісні взаємозв'язки. В нашому випадку облікової ДІС такі взаємозв'язки існують між підсистемами «Бухгалтерський облік» та «Податковий облік». Їх існування відображає взаємозв'язки між відповідними розділами обліку в ПрО, що відображено в відповідних нормативних актах України. На підставі опису ПрО формується матриця взаємозв'язків між кластерами ПрО та відповідним ним підсистемам ДІС. В наступній операції 1.3, на підставі визначеної множини підсистем формується матриця зв'язків між цими підсистемами. Для формування матриці проводиться визначення окремих об'єктів таких підсистем та визначається (за ознакою «0» чи «1») наявність чи відсутність зв'язків між об'єктами визначених підсистем. Наприклад, об'єкт підсистеми «Бухгалтерський облік» – «Господарська операція» та підсистеми «Податковий облік» – «Податкові зобов'язання» та «Податковий кредит» мають однозначні зв'язки – кожна операція, що проведена по бухгалтерському обліку, має бути відображена й в обліку податковому.

На етапі 2 з'ясовуються взаємні зв'язки між підсистемами, які будуть визначати передачу управління та даних в спроектованій ДІС. Етап полягає в побудові діаграм потоків даних (ДПД) між підсистемами й моделюванні передачі даних на підставі створених діаграм. При моделюванні визначаються типи та структури даних, що циркулюють між підсистемами майбутньої ДІС. Слід відмітити, що під даними на цьому етапі маються на увазі не тільки елементарні типи, такі, як число, рядок та інші. Між окремими підсистемами можливий обмін більш складно агрегованими даними – колекції та структури різнотипних значень. На етапі 3 виконується подальша деталізація структури. Тепер деталізується внутрішня структура окремих підсистем. Метою цього етапу є визначення складу підсистеми та її структури. Склад підсистеми визначає набір програмних модулів, що виконують основні алгоритми по обробці даних.

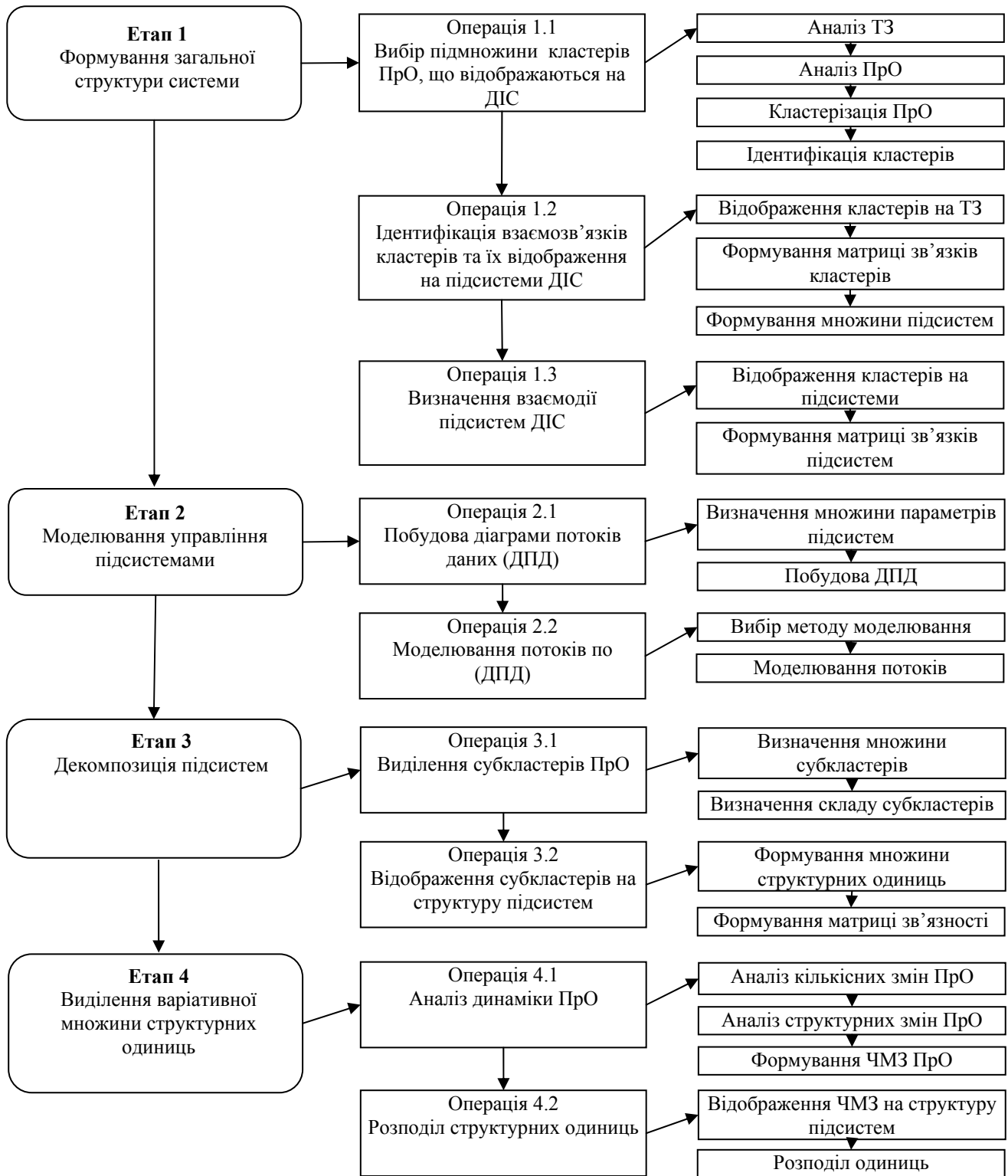


Рис. 2 – Етапи структурного синтезу ДІС

Структуру підсистеми визначають взаємні зв'язки між цими модулями. В операціях 3.1 та 3.2 такі модулі позначено як «субкластери». На етапі 3.2 з'ясовується ступінь зв'язності (СЗ) між окремими модулями підсистеми. Існує сім типів зв'язності [7]:

1. Зв'язність по збігу (СЗ = 0). Відсутні явно виражені внутрішні зв'язки.

2. Логічна зв'язність (СЗ=1). Частина модуля об'єднані за принципом функціональної подібності.

3. Часова зв'язність (СЗ=3). Частина модуля не пов'язані, але потрібні в один і той же період роботи системи.

4. Процедурна зв'язність (СЗ=5). Частина модуля пов'язані порядком виконуваних дій, що

реалізують деякий сценарій поведінки.

5. Комунікативна зв'язність ( $C3=7$ ). Частина модуля пов'язані за даними (працюють з однією і тією ж структурою даних).

6. Інформаційна (послідовна) зв'язність ( $C3=9$ ). Вихідні дані однієї частини використовуються як вхідні в іншій частині модуля.

7. Функціональна зв'язність ( $C3=10$ ). Частина модуля разом реалізують одну функцію.

Для визначення рівня зв'язності модуля (показника  $CC$ ) використовується алгоритм, запропонований в [8]. Він містить наступні кроки:

1. Якщо модуль – одинична проблемно-орієнтована функція, то  $CC=10$ ; кінець алгоритму. Інакше перейти до пункту 2.

2. Якщо дії усередині модуля пов'язані між собою, то перейти до пункту 3. Якщо дії усередині модуля незалежні, то перейти до пункту 6.

3. Якщо дії усередині модуля пов'язані даними, то перейти до пункту 5.

4. Якщо дії усередині модуля пов'язані потоком управління, перейти до пункту 5.

5. Якщо порядок дій усередині модуля важливий, то  $CC = 9$ , інакше  $CC = 7$ . Кінець алгоритму.

6. Якщо порядок дій усередині модуля важливий, то  $CC = 5$ , інакше  $CC = 3$ . Кінець алгоритму.

7. Якщо дії усередині модуля належать до однієї категорії, то  $CC = 1$ . Якщо дії усередині модуля не належать до однієї категорії, то  $CC = 0$ . Кінець алгоритму.

На етапі 4 виконується розподіл множини модулів кожної підсистеми на варіативну та сталу підмножини. Для цього в операції 4.1 виконується частотний аналіз змін ПрО й будується частотна матриця-стовпець змін, яка для кожного модулю показує відносну частоту його змін при змінах ПрО. Відносна частота обчислюється як відношення кількості змін заданого модулю до загальної кількості змін ПрО.

В операції 4.2. власне й виконується розподіл на варіативну та сталу підмножину. До варіативної підмножини виносяться алгоритми, що мають найменший ступінь зв'язності та найбільшу частоту змін.

Слід відмітити, що більшість етапів запропонованого методу не можуть бути формалізовані. Тому побудова автоматичної системи структурного синтезу ДІС на сьогодні є невирішеною проблемою.

### 3. ПРИКЛАД РЕАЛІЗАЦІЇ МЕТОДУ СТРУКТУРНОГО СИНТЕЗУ ДІС

Запропонований метод структурного синтезу реалізовано при створенні облікової ДІС «АгроКомплекс» для автоматизації усіх видів обліку на сільськогосподарських підприємствах. Облік у сільському господарстві має дуже складну ПрО, опис якої наводиться як в законодавчих актах України, так і великій кількості підзаконних інструкцій та постанов різних відомств. Це робить автоматизацію обліку у сільському господарстві складною та трудомісткою задачею.

Метою створення здатної до адаптації структури ДІС «АгроКомплекс» була необхідність забезпечити зниження часу простоїв системи при таких змінах ПрО:

- зміна алгоритмів проведення існуючих та створення нових господарських операцій;
- зміна обраної підприємством системи оподаткування;
- зміни статей податкового обліку при проведенні господарських операцій;
- можливість вибору, зміни або створення власних алгоритмів надання пільг при стягненні податкових коштів;

– можливість вибору та зміни існуючих або створення нових правил розмежування доступу до об'єктів та даних інформаційної бази ДІС.

Практична експлуатація ДІС «АгроКомплекс» в дев'яти господарствах Одеської області показала, що побудова ДІС за запропонованою структурою дозволила скоротити час простоїв на оновлення через мережу Інтернет. Так, наприклад, впровадження облікової ДІС «АгроКомплекс» із гнучкою структурою, в якій алгоритми обробки, що змінюються винесено до складу бази даних дозволило в 1,87 рази скоротити час й коефіцієнт простоїв при оновленні через мережу Інтернет та зменшити обсяг інформаційного пакету оновлень з 8,1 Мбайт до 196 Кбайт.

Впровадження ДІС «АгроКомплекс» із гнучкою структурою в сільськогосподарському підприємстві «Агролідер Т-7» (м. Татарбунари Одеської області) із підключенням до мережі Інтернет через звичайну телефонну лінію зв'язку дозволило скоротити коефіцієнт простоїв на оновлення через в 3,2 рази при тому ж зменшенні обсягу інформаційного пакету. Для порівняння показників використано аналогічну за призначенням серійну інформаційну систему «1С:Предприятие 8. Управление сельскохозяйственным предприятием» фірми 1С [12].

#### 4. ВИСНОВКИ

В статті обґрунтовано проблеми та протиріччя, що виникають на етапі експлуатації динамічних інформаційних систем при змінах їх предметної області. Показано, що зменшення часу простоїв при оновленнях ДІС можливе тільки за рахунок скорочення обсягу інформаційного пакету, який передається від розробника ДІС до користувачів. Розроблено концепцію, за якою частина програмного коду ДІС зберігається як текст в базі даних системи, що дозволяє змінювати лише окремі ділянки програмного забезпечення ДІС. Створено метод побудови ДІС із гнучкою, здатною до адаптування при змінах ПрО структурою, використання якої призводить до зменшення часу простоїв та забезпечення, за рахунок цього, надійності при їх експлуатації; на базі створеної концепції розроблено методи побудови ДІС із гнучкою структурою для забезпечення надійності при їх оновленнях на етапі експлуатації.

Їх використання дозволило розв'язати протиріччя між потребою зменшення простоїв ДІС на етапі експлуатації та необхідністю тривалого призупинення їх роботи для виконання оновлень при змінах ПрО, а також протиріччя між необхідністю одночасної передачі великих за обсягом інформаційних пакетів для оновлення ДІС до великої кількості користувачів, та обмеженою пропускнуою здатністю каналів зв'язку.

Практичне впровадження ДІС із запропонованою структурою довело можливість збільшення надійності ДІС на етапі експлуатації за рахунок зниження часу простоїв при оновленнях.

#### 5. СПИСОК ЛІТЕРАТУРИ

- [1] D. A. Maevsky, H. J. Maevskaya, V. N. Antoschuk, Adaptation of functioning of the accounting information systems to the requirements normatively-legal acts, *Electrical machine-building and electrical equipment*, 72 (2009), pp. 153-160. (in Ukrainian)
- [2] V. N. Petrov, *Information Systems*, SPb: Piter, 2003, 688 p. (in Russian)
- [3] V. A. Matiash, A. V. Nikandrov, V. A. Putilov, A. E. Fedorov, V. V. Filchakov, *Structural Analysis at Software of the Real-Time Systems Development*, Apatity, KF PetrSU, 1997, 78 p. (in Russian)
- [4] M. G. Radchenko, E. J. Hrustaliova, *Architecture and Work with Data «1С:Enterprise 8.2»*, Moscow, 1С Publishing, 2010, 268 p. (in Russian)

- [5] D. A. Maevsky, T. J. Tintulova, V. N. Antoschuk, The information system agricultural «Complex» for a book-keeping and operative account in agriculture, *Agrarian announcer of Black sea. Engineering sciences*, 48 (2009), pp. 151-156. (in Ukrainian)
- [6] D. A. Maevsky, V. N. Antoschuk, Methodology of construction of the information systems is with the variable algorithm of treatment of information, *Works of the Luhansk branch of the International academy of Automation*, 19 (2009), pp. 111-112. (in Russian)
- [7] V. Chistiakov, Modern facilities of development: comparison of productivity, *Technology «Client-server»*, 3 (2001), pp. 18-48. (in Russian)
- [8] V. Chistiakov, Who is most rapid today? <http://www.rsdn.ru/article/devtools/perftest.xml>. (in Russian)
- [9] M. Guillaume, The speed, size and dependability of programming languages, <http://blog.gmarceau.qc.ca/2009/05/speed-size-and-dependability-of.html>. – 2009. (in English)
- [10] S. Amjan, R. K. Reddy, B. Manda, C. Prakashini, K. Deepthi, An empirical validation of object-oriented design metrics for fault prediction, *Journal of Computer Science*, (7) 4 (2008), pp. 571-577. (in English)
- [11] Bases of planning of the programming systems. <http://www-csd.univer.kharkov.ua/content/files/cat16/opps.pdf>. (in Russian)
- [12] 1С:Enterprize 8. Management by an agricultural enterprise, <http://solutions.1c.ru/catalog/agr-enterprise>. (in Russian)



**Антощук Світлана Григорівна**, директор ІКС ОНПУ, завідувача кафедрою інформаційних систем, д.т.н., професор. Після закінчення в 1981 році Одеського політехнічного інституту (зараз – ОНПУ) працювала в ньому. Пройшла шлях від інженера

та асистента до професора. В 2005 році одержала наукову ступінь доктора технічних наук та вчене звання професора кафедри інформаційних систем. Нагороджена нагрудними знаками МОНУ «Відмінник освіти України» та «За наукові досягнення» та грамотами Міністерства освіти і науки України. Автор більш 150 наукових та науково-методичних робіт

Сфера наукових інтересів – автоматизовані управляючі системи з обробкою візуальної

інформації; інформаційні системи; системи штучного інтелекту.



**Масєвський Дмитро Андрі-йович**, завідувач кафед-рою теоретичних основ та загальної електротехніки ОНПУ, к.т.н., доцент. В 1973 поступив, а в 1978 році з відзнакою закінчив Оде-ський політехнічний інсти-тут (зараз – ОНПУ). Працював на кафедрі тео-

ретичних основ електротехніки (нині – теоре-тичних основ та загальної електротехніки) на

посадах асистента, старшого викладача, до-цента. В 1986 році захистив кандидатську дисертацію, присвячену розробці методів проектування друкованих плат обчислювальної техніки надвисокої швидкодії. В 1991 році отримав звання доцента. З 2005 року очолює кафедру.

Сфера наукових інтересів – надійність програмного забезпечення, перехідні процеси в полоскових лініях зв'язку. Опублікував 46 наукових праць та 20 методичних вказівок.





## STRUCTURAL SYNTHESIS OF DYNAMIC INFORMATION SYSTEMS

Svetlana Antoschuk, Dmitry Maevsky

Odessa National Polytechnic University  
1, Shevchenko boulevard, 65044, Odessa, Ukraine  
e-mail: asg@ics.opu.ua, Dmitry.A.Maevsky@gmail.com

**Abstract:** *Theoretical bases are considered and the method of synthesis of the dynamic information systems structure is created with a capacity for adaptation at the changes of a subject domain. A capacity for adaptation is provided due to the selection of a variation part of algorithms of the system that changing at the changing the subject domain. The variation part of algorithms is kept in the database of the system that provides their rapid replacement and promotes a fail safety on the stage of exploitation.*

**Keywords:** *dynamic information system, synthesis of structure, variant algorithms, reliability.*

### 1. INTRODUCTION

The article is dedicated to the description of the method of synthesis of the structure of dynamic information systems (DIS). Dynamic information systems mean information systems subject domain (SD) of which changes at the stage of exploitation. At present time DIS are most widely used systems. At the same time, there is a necessity of frequent alteration of DIS which leads to temporary interruption of its work. Timeouts of DIS which happen during the alterations reduce their reliability, may lead to interruption of industrial processes and significant material losses. Retention of the reliability is only possible through the timeout reduction during the alterations.

Retention of the reliability may be made possible by means of a specially designed DIS structure which allows adaptation of the system to the varying requirements of the SD.

### 2. MAIN CONCEPT

Basic concept of designing a DIS structure which allows fast adaptation for changed requirements of SD is in the selection of a variable subset of algorithms, which change during the alterations of SD, from the multitude of algorithms for data processing. It is suggested to save the program codes of variable algorithms in the system database (DB) in the form of source or partly compiled code («p-code»).

During DIS updates it is enough to alter one or several variable algorithms in the DB. Main program

DIS modules will not be changed which allows sending smaller update data packets to the users.

### 3. DIS STRUCTURAL SYNTHESIS METHOD

Operation of DIS with the suggested structure is illustrated in the fig. 1.

User's query is analyzed by the managing subsystem for determining the subset of the processing algorithms.

If the algorithm is variable, it is extracted from the DIS DB and is handed over to the interpreter. Constant algorithms are carried out immediately.

To design the structure of DIS which allows adaptation for changed requirements of SD a method of structural synthesis was developed (fig.2).

Its objective is to form a variable subset of algorithms.

Variable algorithms refer to DIS algorithms which have the least connectivity with others and are the most changed.

### 4. EXAMPLE OF STRUCTURAL SYNTHESIS METHOD

Suggested method of structural synthesis is realized in the account DIS «Agrocomplex» for automation of all kinds of accounting at the agricultural enterprises.

Practical exploitation of DIS «Agrocomplex» at 9 enterprises of Odessa region has showed that designing a DIS with the suggested structure

allowed reducing the coefficient of timeouts from 1,8 up to 3,2 times as well as diminishing the size of the updating data packet from 8,1 MB up to 196 KB.

## 5. CONCLUSIONS

A concept and method of DIS design with a flexible and adaptable at the changes of SD structure is presented in the article. It allowed reducing of timeouts and increasing of these systems reliability.

This approach made it possible to solve the controversy between the necessity of system program updating and reducing the timeouts of DIS. Reducing the size of data packets and decreasing of requirements for the network equipment are the advantages of the suggested system.

## 6. REFERENCES

- [1] D. A. Maevsky, H. J. Maevskaya, V. N. Antoschuk, Adaptation of functioning of the accounting information systems to the requirements normatively-legal acts, *Electrical machine-building and electrical equipment*, 72 (2009), pp. 153-160. (in Ukrainian)
- [2] V. N. Petrov, *Information Systems*, SPb: Piter, 2003, 688 p. (in Russian)
- [3] V. A. Matiash, A. V. Nikandrov, V. A. Putilov, A. E. Fedorov, V. V. Filchakov, *Structural Analysis at Software of the Real-Time Systems Development*, Apatity, KF PetrSU, 1997, 78 p. (in Russian)
- [4] M. G. Radchenko, E. J. Hrustaliova, *Architecture and Work with Data «1C:Enterprise 8.2»*, Moscow, 1C Publishing, 2010, 268 p. (in Russian)
- [5] D. A. Maevsky, T. J. Tintulova, V. N. Antoschuk, The information system agricultural «Complex» for a book-keeping and operative account in agriculture, *Agrarian announcer of Black sea. Engineering sciences*, 48 (2009), pp. 151-156. (in Ukrainian)
- [6] D. A. Maevsky, V. N. Antoschuk, Methodology of construction of the information systems is with the variable algorithm of treatment of information, *Works of the Luhansk branch of the International academy of Automation*, 19 (2009), pp. 111-112. (in Russian)
- [7] V. Chistiakov, Modern facilities of development: comparison of productivity, *Technology «Client-server»*, 3 (2001), pp. 18-48. (in Russian)
- [8] V. Chistiakov, Who is most rapid today? <http://www.rsdn.ru/article/devtools/perftest.xml>. (in Russian)
- [9] M. Guillaume, The speed, size and dependability of programming languages, <http://blog.gmarceau.qc.ca/2009/05/speed-size-and-dependability-of.html>. – 2009. (in English)
- [10] S. Amjan, R. K. Reddy, B. Manda, C. Prakashini, K. Deepthi, An empirical validation of object-oriented design metrics for fault prediction, *Journal of Computer Science*, (7) 4 (2008), pp. 571-577. (in English)
- [11] Bases of planning of the programming systems. <http://www-csd.univer.kharkov.ua/content/files/cat16/opps.pdf>. (in Russian)
- [12] 1C:Enterprise 8. Management by an agricultural enterprise, <http://solutions.1c.ru/catalog/agr-enterprise>. (in Russian)