

ТЕХНОЛОГІЇ РОЗРОБКИ ТА СУПРОВОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
УДК 004.413(045)

Сидоров М.О.

Національний авіаційний університет

МЕТОДИ І ЗАСОБИ СТВОРЕННЯ І СУПРОВОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕЛИКИХ РОЗПОДІЛЕНИХ КОМП'ЮТЕРИЗОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Розглядається розв'язання актуального науково-практичного завдання створення і супроводження програмного забезпечення великих розподілених комп'ютеризованих інформаційних систем, в тому числі авіаційного призначення.

Концепція системи систем спрямована на застосування успадкованих систем на основі високопродуктивної ІТ – інфраструктури, а відрізняється від традиційних монолітних систем як загальними властивостями (динамічне адаптування до несподіваних ситуацій, продовження розвитку і функціонування шляхом еволюції), так і властивостями, які характеризують системи систем (експлуатаційна незалежність, незалежність за керуванням, несподівана поведінка, географічна розподіленість). Застосування концепції системи систем забезпечує високу оперативність реалізації програмного забезпечення великої комп'ютеризованої системи завдяки застосуванню успадкованих систем, а використання концепції ІТ – інфраструктури – застосування існуючої комп'ютерно-телекомунікаційної структури.

Таким чином створюються умови щодо збереження накопиченого досвіду шляхом об'єднання, розвитку та комплексного застосування успадкованих комп'ютеризованих систем і технологій як в апаратному так і в інформаційно-програмному аспектах та надається поштовх створенню нових систем, які забезпечать ефективне управління.

Представлені методи та засоби, які протягом декількох років створювалися на кафедрі інженерії програмного забезпечення Національного авіаційного університету і спрямовані на конструктивне створення і супроводження програмного забезпечення великих розподілених комп'ютеризованих інформаційних систем.

Рассматривается решение актуальной научно-практической задачи создания и сопровождения программного обеспечения больших распределенных компьютеризованных информационных систем, в том числе авиационного назначения.

Концепция системы систем направлена на применение унаследованных систем на основе высокопроизводительной ИТ - инфраструктуры, а отличается от традиционных монолитных систем как общими свойствами (динамическая адаптация к неожиданным ситуациям, продолжение развития и функционирования путем эволюции), так и свойствами, которые характеризуют системы систем (эксплуатационная независимость, независимость по управлению, неожиданное поведение, географическая распределенность). Применение концепции системы систем обеспечивает высокую оперативность реализации программного обеспечения большой компьютеризованной системы благодаря применению унаследованных систем, а использование концепции ИТ - инфраструктуры - применение существующей компьютерно-телекоммуникационной структуры. Таким образом создаются условия для сохранения накопленного опыта путем объединения, развития и комплексного применения унаследованных компьютеризованных систем и технологий как в аппаратном так и в информационно-программном аспектах и для создания новых систем, которые обеспечат эффективное управление.

Представлены методы и средства, которые в течение нескольких лет создавались на кафедре инженерии программного обеспечения Национального авиационного университета и направлены на конструктивное создание и сопровождение программного обеспечения больших распределенных компьютеризованных информационных систем.

We consider the solution of current scientific and practical task creating and maintaining software of large distributed computer information systems, including aviation systems.

The concept of the system of systems aims to use legacy systems based on high-performance IT - Infrastructure and differs from traditional monolithic systems as general properties (dynamic adaptation to unexpected situations, continuing development through evolution) and the properties that characterize the system of systems (operational independence, independence for running, unexpected behavior, global geographic). Application of the concept of the system of systems provides high speed software implementation of large computerized information system through utilizing legacy software and using the concept of IT infrastructure. Thus, the conditions for the preservation of experience by combining the development and application of complex legacy computer systems and technology in hardware and in software aspects are provided.

Presented methods and tools that were created over several years at the Department of Software Engineering of National Aviation University.

Ключові слова: програмне забезпечення, система систем, авіаційна система, процеси програмного забезпечення.

Вступ

Розглядається розв'язання актуального науково-практичного завдання створення і супроводження програмного забезпечення великих розподілених комп'ютеризованих інформаційних систем, в тому числі авіаційного призначення.

Стан розв'язання завдання характеризується первинною наявністю окремих успадкованих монолітних комп'ютеризованих систем і відповідних технологій побудови їх програмного забезпечення. Тому, для реалізації великих розподілених комп'ютеризованих систем запропоновано підхід на основі застосування концепції системи систем (system of systems), яка враховує цей стан і забезпечить майбутній розвиток галузі [1].

Концепція системи систем спрямована на застосування успадкованих систем на основі високопродуктивної ІТ – інфраструктури, а відрізняється від традиційних монолітних систем як загальними властивостями (динамічне адаптування до несподіваних ситуацій, продовження розвитку і функціонування шляхом еволюції), так і властивостями, які характеризують системи систем (експлуатаційна незалежність, незалежність за керуванням, несподівана поведінка, географічна розподіленість). Застосування концепції системи систем забезпечує високу оперативність реалізації програмного забезпечення великої комп'ютеризованої системі завдяки застосуванню успадкованих систем, а використання концепції ІТ – інфраструктури – застосування існуючої комп'ютерно-телекомунікаційної структури [2]. Таким чином створюються умови щодо збереження накопиченого досвіду шляхом об'єднання, розвитку та комплексного застосування успадкованих комп'ютеризованих систем і технологій як в апаратному так і в інформаційно-програмному аспектах та надається поштовх створенню нових систем, які забезпечать ефективне управління. Методи й засоби, що розроблені й досліджені у роботі, можуть також

використовуватись у практичному програмуванні, при масовому виробництві й супроводженні програм.

Стаття містить три розділи. В першому розділі, розглядаються основні положення контексту в якому розроблялися методи і засоби. В другому розділі, наведено конструктивний підхід до створення програмного забезпечення великих, розподілених комп'ютеризованих систем шляхом застосування концепції «системи систем». В третьому розділі, надано стислий опис методів і засобів створення і супроводження програмного забезпечення, що розроблені під керівництвом автора на кафедрі інженерії програмного забезпечення Національного авіаційного університету.

Основні положення

Методи та засоби, що розглядаються, створювалися в контексті інформатизації науково-технічного супроводу технічної експлуатації авіаційної техніки.

Під інформатизацією науково-технічного супроводу технічної експлуатації авіаційної техніки розумітимуться політики і процеси, що спрямовані на організацію продуктивної діяльності людини в даній області шляхом використання системно об'єднаних програмних, інформаційних, комп'ютерних і телекомунікаційних засобів.

Інформатизація вимагає вирішення завдань широкого спектру, від науково-технічних до соціальних. Останнє пояснюється тим, що впровадження інформатизації пов'язано із значними змінами в житті і діяльності людини, яка приймає участь у відповідній господарській діяльності.

Так як, ефективний супровід експлуатації авіаційної техніки неможливий без автоматизованого інформаційного забезпечення за допомогою інформаційних систем, основою яких є комп'ютеризовані системи, а програмне забезпечення – ключова складова інформаційних і комп'ютеризованих авіаційних систем і технологій в тому числі авіаційних, то розв'язання завдання створення і супроводження

програмного забезпечення, є ключовою складовою ефективною реалізацією науково-технічного супроводу експлуатації авіаційної техніки.

Задачі, на розв'язання яких було спрямовано роботу:

– забезпечення методологічних засад, щодо створення і застосування принципів, методів, методик розробки, реалізації та супроводження математичного, інформаційного, програмного, технічного, організаційного та методичного забезпечення інформаційних технологій супроводження експлуатації авіаційної техніки;

– впровадження результатів, в наукові дослідження і зокрема в дослідження програмного забезпечення, навчальний процес та розробку відповідних засобів інформатизації процесів в цьому та інших доменах.

Підхід, який пропонується застосовувати для розв'язання задач полягає в тому, що інформація щодо супроводження експлуатації авіаційної техніки розглядається як результат виконання сукупності процесів і спрямована на задоволення інформаційних потреб супроводження експлуатації авіаційної техніки шляхом розробки, використання і розвитку комплексу масштабуємих безпаперових інформаційно-програмних засобів, заснованих на сучасному комп'ютерному та телекомунікаційному обладнанні.

Розроблювана методологія інформатизації повинна створювати умови щодо збереження накопиченого досвіду шляхом об'єднання, розвитку та комплексного застосування успадкованих інформаційних систем і технологій та надати поштовх створенню нових, які забезпечать ефективне супроводження експлуатації авіаційної техніки.

Розроблена і досліджена методологія ґрунтується на чотирьох принципах: успадкованість, бережливість, інженерність, децентралізованість.

Успадкованість. Реалізація принципу успадкованості досягається завдяки утилізації. Одне з основних завдань утилізації програмного забезпечення полягає в створенні компонентів повторного використання [3,4]. До компонентів повторного використання пред'являються вимоги, для задоволення яких при їх створенні недостатньо досвіду окремого програміста або розгляду окремих застосувань, а необхідно ґрунтуватися на обширнішому матеріалі, застосовуючи спеціальні методи і засоби. Окрім цього, щоб ефективно реалізовувати процеси

пов'язані з відновленням і переробкою програмного забезпечення необхідно розробляти класифікації, плани і трансформації, які б також відповідали вимогам не одного застосування. Тому утилізація спрямована перш за все на досліджування ґносеологічних принципів домену [5].

Бережливість. Процес розробки програмного забезпечення – ресурсномісткий. Розробка програмного продукту може тривати роками, навіть при використанні великого колективу розробників. Готові програмні продукти можуть стати важливою частиною промислових і військових систем, де до них будуть застосовуватися серйозні вимоги, особливо у плані безпеки. Програмні продукти, орієнтовані на широкого споживача, можуть принести компанії-розробнику, як значні прибутки, так і значні збитки. Складання кошторису та оцінка часу необхідного для розробки програмного забезпечення ускладнені внаслідок самої природи процесу. Більшість проблем асоційованих з розробкою програмного забезпечення є проблемами організаційними, які часто призводять до перевитрати коштів, відставання від графіка, випуску неякісного продукту і багатьом іншим проблемам.

Правильна організація процесу розробки і підвищення його ефективності – предмет вивчення багатьох дослідників, особливо зі спільноти agile. Концепція ощадливого виробництва, що розроблена компанією Toyota, з плином часу стала набирати популярність і в інших галузях економіки. З'явилася така «ощадлива» концепція і в інженерії програмного забезпечення [6].

Інженерність. Часи, коли створення програмного забезпечення було мистецтвом минули. Зараз розробка і супроводження програмного забезпечення – це масове виробництво, яке потребує відповідного підходу до його організації. Найголовнішим аспектом цієї організації є інженерія – тобто творче застосування наукових результатів до створення корисних для людини об'єктів, насамперед технічних. Поява галузі – інженерія програмного забезпечення, 50 років тому, свідчить про важливість застосування інженерії в виробництві програмних продуктів [7]. Тому, принцип інженерії забезпечує як ґрунтовну реалізацію процесів життєвого циклу, так і конструктивну основу для проектування і конструювання програмного забезпечення.

Децентралізація і різноманітність. В останні кілька років застосовується новий підхід

щодо дослідження програмного забезпечення. Сутність його полягає у тому, що програмне забезпечення не розглядається на самоті, а його властивості досліджуються в контексті інших об'єктів і суб'єктів, які діють навколо програмного забезпечення. Такий погляд на програмне забезпечення привів до застосування в інженерії програмного забезпечення поняття екосистем [8]. Таким чином поняття екосистема програмного забезпечення та супутні терміни активно використовуються виробниками і дослідниками, позначаючи ним системи, що включають підприємство розробника, його програмного забезпечення і партнерів.

Програмне забезпечення великої комп'ютеризованої системи як система систем

Застосування та дослідження побудови програмного забезпечення великих комп'ютеризованих систем управління як системи систем реалізується на основі сервіс-орієнтованої архітектури [9]). Система систем визначається як множина систем, які взаємодіють для забезпечення визначеної мети [1]. Системи систем мають унікальні характеристики, які відрізняють їх від традиційних монолітних систем, а при функціонуванні залежать від розподіленого управління, співробітництва, впливу, каскадних ефектів та іншої емергентної поведінки як головних композиційних механізмів.

При екосистемному дослідженні програмного забезпечення великої розподіленої комп'ютеризованої інформаційної системи як системи систем, її компонентами будемо вважати наступні: програмне забезпечення (технічна абстрактна система S_1); апаратне забезпечення (технічна матеріальна система S_2); природні об'єкти (природна система S_3); держава, органи стандартизації, виробники та продавці, користувачі (економічна система S_4); розробники та менеджери, соціальні спільноти (соціальна система S_5).

Таким чином, під великою розподіленою комп'ютеризованою системою в цілому будемо розуміти систему систем, утворену взаємодією автономних відкритих систем технічного, природного, соціального та економічного походження з метою забезпечення визначеної продуктивності. В залежності від конкретного аспекту, який цікавить дослідника, автономні компоненти можуть бути доповнені або уточнені.

В програмному забезпеченні реалізуючи принцип інженерії важливо вміти конструктивно

представляти програмні системи. Використовуючи алгебру процесів реального часу (Real Time Process Algebra [10]) можна формально описати поведінку програмної системи кінцевою множиною метаповедінок. Метаповедінки моделюються за допомогою метапроцесів і їх композицій, а також алгебраїчних відносин [11]. Метод, запропонований для вирішення завдання представлення програмної системи як композиції систем-капсул, дозволяє формально описати будь-яке програмне забезпечення в термінах відкритих систем і досліджувати його властивості [12].

Більшість практичних систем реального світу є відкритими. Вони повинні взаємодіяти із зовнішнім світом (середовище Θ) для обміну енергією, речовиною та/або інформацією. Типовими провідниками взаємодії відкритої системи з середовищем є вхід і вихід. Будемо описувати капсулу як кортеж наступного вигляду:

$$S = (C, R^c, R^i, R^0, B, \Omega, \theta),$$

де Θ – зовнішнє середовище з непустих мнжинних компонентів $C_\theta, C_\theta \cap C = \emptyset$;

$R^c \subseteq C \times C$ – множина внутрішніх відносин;

$R^i \subseteq C_\theta \times C$ – множина зовнішніх вхідних відносин;

$R^0 \subseteq C \times C_\theta$ – множина внутрішніх вихідних відносин.

B – множина поведень (або функцій) $B = \{b_1, b_2, \dots, b_p\}$; Ω – множина структур на компонентах, умови відносин, рамки (область дії) поведінки $\Omega = \{\omega_1, \omega_2, \dots, \omega_p\}$.

Композиція систем є найбільш складною операцією і можлива лише між відкритими системами. Вона інтегрує дві або більше систем в суперсистему з ієрархічною архітектурою.

Використовуючи поняття капсул та алгебру систем, можна представити програму як композицію капсул таким чином: $S = S^* \gg \rightarrow S_3 = (S_{31} || S_{32}) \gg \rightarrow S_2 = (S_{20} || S_{21} || S_{22} || S_{23}) \gg \rightarrow S_1$, де $||$ – відношення паралельності, яке використовується для опису композиції між функціонально паралельними або еквівалентними системами; \gg – відношення вкладеності, яке визначає одну систему як вбудовану частину іншої; $S^* = S_4 \vee S_5$, где: S_5 – система «Клас», що описує капсулу класного рівня інкапсуляції; S_4 – система «Модуль», що описує капсулу модульного рівня інкапсуляції; S_3 – система «Підпрограма», яка представлена двома системами S_{31} і S_{32} (відкритою і закритою підпрограмами відповідно), описує капсулу підпрограмного рівня інкапсуляції; S_2 – система «Оператор», що описує капсулу операторного

рівня інкапсуляції, складається з чотирьох підсистем типу «Оператор», а саме: S_{20} – оператор привласнення; S_{21} , S_{22} і S_{23} – операторний базис (оператори послідовності, вибору і повторення відповідно); S_1 – система «Лексема», що описує капсулу лексичного рівня інкапсуляції.

Формально описуватимемо кожну капсулу як систему, слідуючи схемі, що складається з шести дій.

1. Представити систему у вигляді кортежу:

$$S = (C, R^0, R^0, B, \Omega, \theta),$$

2. Формально описати кожен елемент кортежу, застосовуючи:

– для C , θ – теорію множин і операції алгебри над ними;

– для R – теорію відносин, обов'язково перевіряючи виконання властивостей асиметричності і рефлексивності, оскільки у разі невиконання хоч би одного з властивостей кортеж не можна вважати системою;

– для B , Ω – логіку висловів і алгебру процесів реального часу.

3. Врахувати те, що поведінка програмної системи є функцією від обчислювальних операцій, часу і структур пам'яті.

4. Вибрати і формально описати властивості системи, наприклад, розмір, амплітуда, складність, зв'язність, зчеплення.

5. Класифікувати систему, виходячи з оцінки властивостей (п. 4) і ключових характеристик елементів кортежу.

6. Кожну капсулу, яка має підсистеми, розглянути як систему систем і описати формально за допомогою операцій алгебри систем.

Методи та засоби створення та супроводження програмного забезпечення

Представлено розроблені методи та засоби дослідження, створення та супроводження програмного забезпечення великих розподілених комп'ютеризованих систем управління. Усі запропоновані методи і відповідні засоби поділяються на дві групи:

1. Методи і засоби інформаційного забезпечення процесів створення і супроводу програмного забезпечення;

2. Методи і засоби модернізації інформаційного і програмного забезпечення.

Далі наведено стислий опис методів та засобів.

Методи і засоби інформаційного забезпечення процесів створення і супроводу програмного забезпечення.

До цієї групи відносяться методи і засоби, які забезпечують процеси життєвого циклу програмного забезпечення.

Метод та засіб моніторингу дефектів проектування об'єктно-орієнтованого програмного забезпечення. Сутність методу полягає в спостереженні за зміною параметрів дефектів проектування [13]. Метод реалізується шляхом використання запропонованої метамоделі історії дефектів проектування (Design Flaws History Model - DDHM) об'єктно-орієнтованого програмного забезпечення і багатоаспектної візуалізації дефектів проектування уражених елементів конструкції різного рівня абстракції. У DDHM дефект уперше моделюється як окрема сутність, здатна змінювати свої параметри з плином часу. Модель дефекту проектування та методика її побудови дозволяє отримати значення таких параметрів дефекту: ступінь розвитку дефекту, інтенсивності ознак дефекту і середню інтенсивність ознак дефекту.

Методи та засоби експертного оцінювання властивостей повторно використовуваних компонентів програмного забезпечення. Сутністю повторного використання є застосування компонентів існуючого програмного забезпечення під час створення нового програмного забезпечення [14]. Повторне використання визначено одним з найперспективніших шляхів підвищення ефективності створення програмного забезпечення, поряд з удосконаленням організаційно-технічного забезпечення розробки програмного забезпечення, підвищенням ефективності фаз життєвого циклу, усуненням переробки програмного забезпечення. Вперше запропоновано метод визначення множини властивостей компонентів програмного забезпечення, сутєвих з погляду повторного використання, шляхом аналізу властивостей компонентів програмного забезпечення із застосуванням 3С – моделі компонентів програмного забезпечення та з урахуванням впливу властивостей на витрати для виконання процесів повторного використання, які виникають під час розробки програмного забезпечення.

Метод побудови засобів доменного аналізу на основі формальних специфікацій у RTPA. Сутність пропонованого методу полягає в тому, що на основі формалізованого опису процесів доменного аналізу, створеного за допомогою інструментального програмного забезпечення будуються засоби доменного аналізу, спрямовані

на його автоматизоване виконання з метою вирішення конкретного завдання, що виникло перед або в процесі розробки програмного забезпечення [15]. Формалізований опис процесів здійснюється за допомогою RTPA.

Метод та засоби декомпіляції похідного коду. Засоби відновлення інформації про програмне забезпечення – інструменти зворотної інженерії, які призначені для аналізу початкового коду на мові успадкованого програмного забезпечення і побудови його високорівневого алгоритмічного уявлення. Звичайно, ці засоби будуються на основі широко відомої схеми, основу якої складає декомпілятор. Від правильної та ефектної роботи декомпілятора залежить результат процесу реінженерії в цілому. Сутність запропонованого методу полягає в застосуванні шаблону, який шляхом його заміни забезпечує налаштування декомпілятора як по входу так і по виходу (апаратна платформа, мова програмування) [16].

Метод та засіб відновлення інформації про технічне обладнання. Процеси побудови цифрового пристрою зв'язку з об'єктом при реінженерії відповідають процесам реінженерії успадкованого програмного забезпечення і поділяються на дві фази.

Перша – зворотня інженерія спрямована на отримання інформації необхідної для побудови пристрою зв'язку з об'єктом.

Друга – пряма інженерія і спрямована на конфігурування, налаштування і відладку нового пристрою зв'язку з об'єктом. В результаті виконання першої фази виділяється це – набір вхідних і вихідних параметрів і їх характеристики а також характеристики датчиків, приладів і виконавчих пристроїв тренажера. В запропонованому методі йдеться про останній тип інформації, що стосується приладів. Інформація цього типу може бути отримана наступними способами: аналізом існуючої документації; створенням високорівневих представлень успадкованого програмного забезпечення; експериментальним дослідженням поведінки компонентів тренажера. Саме для реалізації такого дослідження пропонується метод, сутність якого полягає в використанні шаблону, а у його складі спеціального набору інструментів вимірників і генераторів сигналів [14].

Методи і засоби модернізації інформаційного і програмного забезпечення.

Представлено розроблені методи, які спрямовані на модернізацію програмного забезпечення авіаційної техніки, що

експлуатується. Застосування запропонованих методів і засобів демонструється на прикладах різних типів авіаційної техніки, від тренажерів до систем забезпечення безпеки польотів.

Методи та засоби редокументування успадкованого програмного забезпечення. У контексті реінженерії успадкованого програмного забезпечення застосовують зворотну інженерію та редокументування. Редокументування застосовується при реінженерії успадкованого програмного забезпечення, коли інформація про програмне забезпечення загублена, є неповною, неточною або неструктурованою. Редокументування визначається як створення документації успадкованого програмного забезпечення на основі відновленої у зворотній інженерії інформації [17]. Використано модельно-керована розробка програмного забезпечення сутність якої полягає у представленні розробки програмного забезпечення як перетворення узгоджених моделей, описуючих програмне забезпечення на різних рівнях абстракції. Для застосування підходу в реалізації методу досліджено концепцію документа програмного забезпечення та його складові з точки зору редокументування на рівнях загальної концепції документу програмного забезпечення, вимог технологій розроблення, вимог до певного документу обраної технології, на основі чого побудовано модель документу програмного забезпечення, мета-модель документу технологій розробки, модель документу технологій розробки і відповідні перетворення вказаних моделей.

Методи та інформаційно-програмні засоби синтезу та аналізу стилів програмування. Для реалізації засобів, що автоматизують вирішення задач формування і застосування стилю програмування, наведено наступне [18]: опис стилю в програмуванні, визначено задачі формування і застосування стилю програмування; запропоновано методи для вирішення поставлених задач та описано процеси, що реалізують відповідні методи; розроблено модель предметної області, яка представлена схемою систематизації правил стилів програмування і формою представлення правил стилю програмування і стилю мови програмування; розроблено математичні моделі пристроїв для вирішення задач формування і застосування стилю програмування; запропоновано використання вимірювань у програмному забезпеченні для вирішення задач

застосування стилю методами контролю і аналізу стилю програми.

Метод створення програмного забезпечення пульта інструктора авіаційного тренажеру.

Використання авіаційних тренажерів для зменшення вартості льотної підготовки, покращення екології набуває особливої актуальності в умовах економічної кризи країни. В зв'язку з цим, зараз до авіаційних тренажерів висуваються більш жорсткі вимоги щодо гнучкості, мобільності та вартісних складових. Невід'ємною частиною авіаційного тренажера є пульт інструктора, який відображає інформацію про політ та стан тренажеру. Існуючі пульти більшості успадкованих тренажерів реалізовані апаратно, морально застарілі та фізично зношені. Це є причиною заміни апаратного пульта інструктора на нові апаратно-програмні засоби на основі сучасних комп'ютерних технологій. Для вирішення цієї задачі пропонується метод, сутність якого полягає в використанні поняття ролі при формуванні інтерфейсу, представленні інтерфейсних елементів як композиції елементів та шаблонному проектуванні при створенні програмного забезпечення пульта [19]. Як розвиток зазначеного підходу для підвищення гнучкості побудови інтерфейсних елементів та настроювання інтерфейсу під різні види задач пропонується застосування шаблонного проектування і узагальнення інформації на основі поняття ролі.

Метод і засіб управління зручністю використання програмних продуктів. Вирішення задачі управління зручним використанням за розробленим методом передбачає наступні етапи [20]:

1) Побудова експертами ієрархічної структури зручного використання. Включає розробку метрик структурним методом зверху-вниз (IEEE Std. 1061).

2) Обчислення значень показників зручного використання на підставі метрик, значення яких отримуються на основі оцінок користувачів.

3) Побудова математичної моделі оцінки зручного використання.

4) Побудова математичної моделі забезпечення зручного використання. Математична модель оцінки зручного використання доповнюється функцією трудомісткості зміни показників зручного використання, тим самим, отримуючи модель забезпечення встановленого рівня зручного використання оптимальним чином.

5) Формування оптимального варіанту забезпечення встановленого рівня зручного

використання. Результатом буде набір показників, що потребують покращення з урахуванням величини зміни кожного показника.

б) Впровадження отриманого варіанту змін показників та контроль досягнення встановленого рівня зручних використань на наступній ітерації, при необхідності – коригування моделей.

Принциповою особливістю даного методу є те, що в ньому передбачається не лише оцінка, але й управління зручними використаннями в процесі розробки програмного продукту.

Метод керованої об'єктом реінженерії програмного забезпечення авіаційних тренажерів.

Особливості домену та місця тренажеру в ньому визначають те, що реінженерія успадкованого програмного забезпечення цього типу окрім вирішення традиційних завдань, пов'язаних з відновленням проектної інформації, вимагає відновлення інформації про реальний об'єкт та особливого підходу до рішення задачі обґрунтування адекватності функціонування побудованого в результаті реінженерії програмного забезпечення поведінці реального об'єкту [14]. Інформація про реальний об'єкт, відновлювана в процесі реверсивної інженерії, – це алгоритми, моделі, характеристики вхідних параметрів, датчиків, приладів та виконавчих пристроїв або моделі реального об'єкту. При відновлюванні, окрім шляху для отримання інформації – аналізу успадкованого коду та документації – слід використовувати експериментальне дослідження поведінки моделі.

Висновки

Розв'язано актуальне, складне науково-технічне завдання створення і супроводження програмного забезпечення великих розподілених комп'ютеризованих інформаційних систем та отримано наступне:

– розроблено наукові основи, досліджено методи й засоби спрямовані на великих розподільних комп'ютеризованих інформаційних систем шляхом розв'язання задач створення і супроводження програмного забезпечення конструктивного представлення

– викладені результати отримані під керівництвом автора на кафедрі інженерії програмного забезпечення Національного авіаційного університету при виконанні НДР (№ 779 – ДБ12 «Методологія інформації супроводження експлуатації авіаційної техніки, в тому числі безпілотних літальних апаратів», МОНУ) та низки дисертаційних робіт.

Більшість результатів не мають закордонних аналогів.

Список використаних джерел

1. Jamshidi M. System of systems// Jamshidi M.// – Wiley, 2009. – 647р.
2. Теленик С.Ф. Спосіб підвищення ефективності інформаційного забезпечення систем управління повітряним рухом / Теленик С.Ф., Грешин І.Ю.// – Вісник НТУУ «КПІ». Серія «Інформатика і обчислювальна техніка». – Вип. 51. – 2009.
3. Сидоров Н.А. Повторное использование программного обеспечения // Кибернетика. – 1989. – №3. – С. 46 – 50.
4. Сидоров Н.А. Средства хранения поиска повторно используемых программных компонентов/ Сидоров Н.А., Шерепа А.Н. // УСиМ. – 1991. – № 6. – С. 20 – 28.
7. Сидоров Н.А. Восстановление, переработка и повторное использование программного обеспечения / Сидоров Н.А. // УСиМ. – 1998 – № 3. – С. 50-59.
8. Сидоров Н.А. Зеленые информационные системы и технологии./ Сидоров Н.А. // Інженерія програмного забезпечення. – 2011. – №3(7). – С. 5 – 13.
9. Сидоров Н.А. Инженерия программного обеспечения – дисциплина или подготовка бакалавров / Сидоров Н.А.// Управляющие системы и машины. – 2006. – №2. – С. 30 – 36.
10. Сидоров Н.А. Экология программного обеспечения / Сидоров Н.А. // Інженерія програмного забезпечення. – 2010. – №1(1). – С. 53 – 61.
9. SOA Principles of Service Design by Thomas E. – Prentice Hall/Pearson, 2009. – 573 p.
10. Wang Y. Software engineering foundation//Wang Y./ – Auerbach Pub., 2007. – 1006 p.
11. Сидоров Н.А. Системная инженерия программного обеспечения / Сидоров Н.А., Луцкий М.Г., Гученко И.В.// Інженерія

програмного забезпечення. – 2010. – №4. – С. 13 – 25.

12. Сидоров Н.А. Применение принципов программной инженерии в преподавании основ программирования / Сидоров Н.А. //УСиМ. – 1999 г. – № 4. С. 54 – 64.
13. Сидоров М.О. Методи та засоби виявлення дефектів проектування об'єктно-орієнтованого програмного забезпечення / Сидоров М.О. //Вісник НАУ. – К.: НАУ. – 2009. – №3. – С. 200 – 205.
14. Сидоров Н.А. Реинженерия наследуемого программного обеспечения информационно-моделирующих тренажерных комплексов / Сидоров М.О. // УСИМ. – К.: 2008. – №4. – С. 68 – 74.
15. Сидоров М.О. Конструктивный підхід до проектування програмного забезпечення / Сидоров М.О., Гученко І.В. // Вісник Східноукраїнського національного університету імені В. Даля. – 2013. – №12. – С. 67 – 73;
16. Сидоров Є.М. Зворотна інженерія програмного забезпечення авіаційних тренажерів // Інформаційні технології та комп'ютерна інженерія. – К.: НАУ. – 2013. – №2. – С. 33 – 38;
17. Сидоров Н.А. Реинженерия проектов программного обеспечения / Сидоров Н.А., Хоменко В.А., Авраменко О.А //Проблемы программирования. – К.: НАУ. – № 2, 3. – 2006. – С.282 – 289.
18. Сидоров М.О. Software stylistics / Сидоров М.О. //Вісник НАУ. – 2005. – №2 (24). – С.98 – 103.
19. Сидоров М.О. Підтримка придатності та продовження експлуатації програмного забезпечення авіаційної техніки / Сидоров М.О., Луцький М.Г., Рябокін Ю.М. // Проблеми програмування. – К.: НАУ, 2010. – №2,3. – С. 229 – 237.
20. Guchenko I.V. Usability management in the context of software architecture // Інженерія програмного забезпечення. – № 2(18). – 2014. – С. 20 – 25.

Відомості про автора:



Сидоров Микола Олександрович – доктор технічних наук, професор, завідувач кафедри інженерії програмного забезпечення Інституту комп'ютерних інформаційних технологій Національного авіаційного університету. Наукові інтереси: інженерія програмного забезпечення.

E-mail: nikolay.sidorov@livenau.net