

## ПІДТРИМКА ПРИДАТНОСТІ ТА ПРОДОВЖЕННЯ ЕКСПЛУАТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВІАЦІЙНОЇ ТЕХНІКИ

*М.Г. Луцький, М.О. Сидоров, Ю.М. Рябокін*

Національний авіаційний університет. Київ, проспект Комарова, 1, корп. 6,  
[nikolay.sidorov@livenau.net](mailto:nikolay.sidorov@livenau.net), [yulia.ryabokin@livenau.net](mailto:yulia.ryabokin@livenau.net)

Розглядається реінженерія, як шлях щодо підтримки придатності та продовження експлуатації програмного забезпечення авіаційної техніки. На прикладах показується застосування доменного аналізу і зворотної інженерії для отримання рішень, що можна повторно використовувати.

The software reengineering as the way of the maintenance of the of aviation technics software is presented. The two concrete exersaices demonstrate using domain analisys and reverse engineering for realizing the aviation software.

### Вступ

Багато десятиліть в авіаційній техніці застосовуються обчислювальні системи, важливою складовою яких є програмне забезпечення. При експлуатації авіаційної техніки для її програмного забезпечення також як і для інших її складових виникає задача підтримки придатності та продовження експлуатації. Розглядається застосування реінженерії, як шляху до вирішення цієї задачі для програмного забезпечення. Демонструється досвід застосування реінженерії програмного забезпечення у двох випадках та на двох різних прикладах авіаційної техніки – авіаційні тренажери та системи контролю польотів авіації.

### Огляд існуючих досліджень

Задача реінженерії програмного забезпечення авіаційної техніки виникає в двох випадках. По-перше – при здійсненні доробки, модифікації або відновлення авіаційної техніки, по-друге – при створенні нових поколінь авіаційної техніки [1–3]. У першому випадку при вирішенні задачі підтримки придатності програмного забезпечення здійснюється часткова або повна його переробка. В другому – програмне забезпечення створюється заново, але обов'язково з урахуванням попереднього досвіду. В обох випадках застосовують реінженерію програмного забезпечення, яка має власні методи та засоби [4]. В залежності від галузі використання реінженерії, застосування методів і засобів має або особливості, або частіше, потребує розробки нових методів і засобів, звичайно на основі існуючих підходів [5, 6].

Авіаційні інформаційно-моделюючі комплекси, це об'єкти, які містять значну кількість різноманітного обладнання, що працює під управлінням обчислювальних систем [5]. Коли виникає задача підтримки працездатності або продовження експлуатації таких комплексів, обов'язкові зміни в обладнанні потребують змін в інформаційно-програмному забезпеченні [7]. Шлях реалізації цих змін – застосування реінженерії програмного забезпечення, яка для цих комплексів (на відміну від інших) має певні особливості. Їх сутність полягає в тому, що при виконанні складових реінженерії – доменного аналізу та зворотної інженерії досліджується не тільки успадковане програмне забезпечення, але й технічне обладнання. В зв'язку з цим для виконання зворотної інженерії, як складової реінженерії, необхідно розглядати два типи засобів. Перші застосовуються традиційно, для дослідження успадкованого програмного забезпечення [4], другі, нові засоби, – для досліджень технічного обладнання, що й пропонується в роботі. Результати обох досліджень, використовують для налагодження відновленого програмного забезпечення або для реалізації того програмного забезпечення, що розробляється заново.

Засоби досліджень складаються з методів та інструментів відновлення інформації про успадковане програмне забезпечення, шляхом зворотної інженерії [4, 7] та про рішення, що повторно використовується, з реалізаційного та прикладного доменів, шляхом доменного аналізу [8, 9] (рис. 1).

Доменний аналіз уперше введений для позначення дій, що забезпечують виявлення об'єктів, операцій і відносин з метою їх повторного використання в розробці нових застосувань домену [8]. Існують різні підходи до доменного аналізу. В даній роботі застосовується підхід R.Prieto-Diaz [9]. Авіаційний контекст доменного аналізу показано на рис. 2.

Виконання доменного аналізу включає три послідовні процеси – підготовка інформації, власне в аспекті авіаційної техніки, доменний аналіз, створення повторно використовуваних компонентів [8]. Обов'язковими продуктами доменного аналізу є вимоги до застосування домена, шаблони, таксономія домена, а необов'язковими є модель і мова домена.



Рис. 1. Доменний аналіз та зворотня інженерія

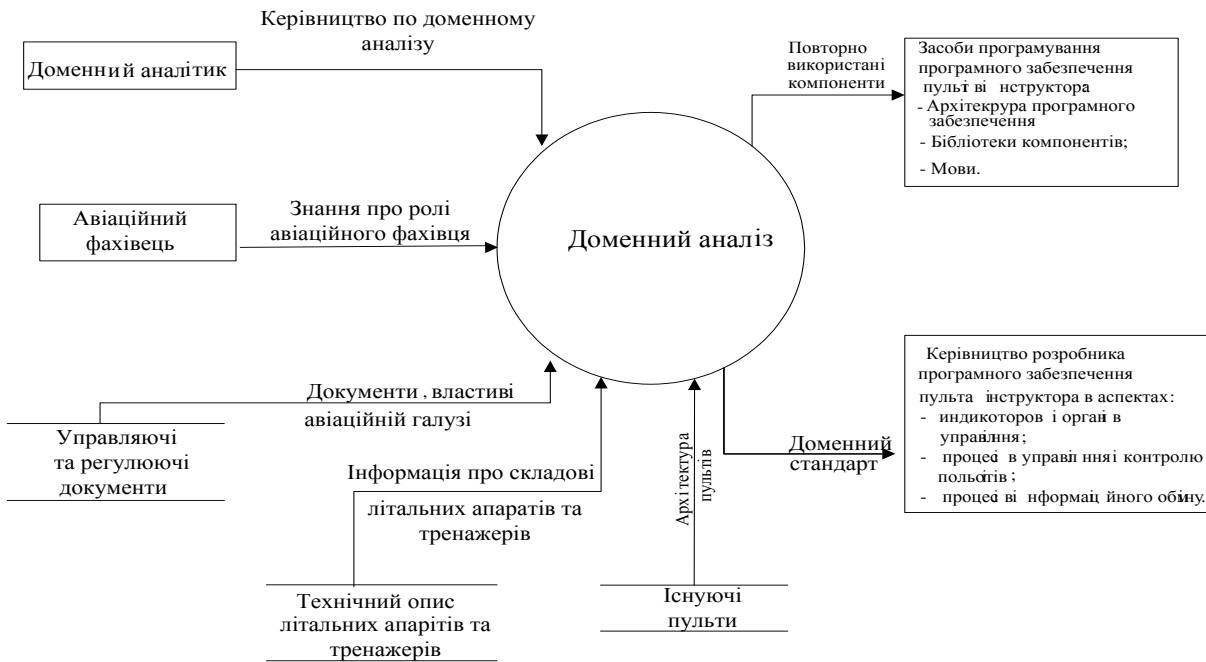


Рис. 2. Контекст доменного аналізу в авіаційному домені

Засоби відновлення інформації про програмне забезпечення, це інструменти зворотної інженерії, які призначені для аналізу початкового коду на мові успадкованого програмного забезпечення й побудови його високорівневого алгоритмічного уявлення. Зазвичай ці засоби будуються на основі досить відомої схеми, яка містить компоненти призначені для здобування відповідної інформації з похідного коду та компонентів, які на основі цієї інформації будують представлення знань про програму на різних рівнях абстрагування (рис. 3) [4].

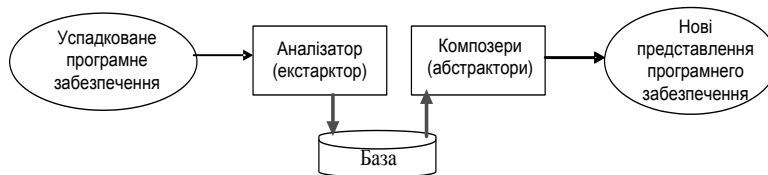


Рис. 3. Схема засобів дослідження програмного забезпечення

### Створення програмного забезпечення пульта інструктора на основі доменного аналізу

Нині для створення сучасних пультів інструкторів використовують підхід «фіксована функціональність», сутність якого полягає в тому, що склад і функціональність пульта визначаються на основі відомих типових задач, які виконуються інструктором при навчанні пілотів [10, 11]. Основним недолік такого підходу – після створення пульта види і форми інформації, що вводиться та відображається не змінюються. Для внесення змін необхідна наявність розробника.

© М.Г. Луцький, М.О. Сидоров, Ю.М. Рябокін, 2010

Для подолання зазначеного недоліку на основі доменного аналізу розробляється рішення про можливість повторного використання. Сутність рішення полягає в тому, що кожен елемент інтерфейсу (інтерфейсний елемент) пульта інструктора тренажера (кнопка, індикатор тощо) реалізується у вигляді окремого програмного модуля, який динамічно підключається та представляє собою .Net-збірку. З окремих інтерфейсних елементів збирається певний інтерфейс.

Відповідний фрагмент доменного аналізу у вигляді діаграми потоків даних показано на рис. 4.

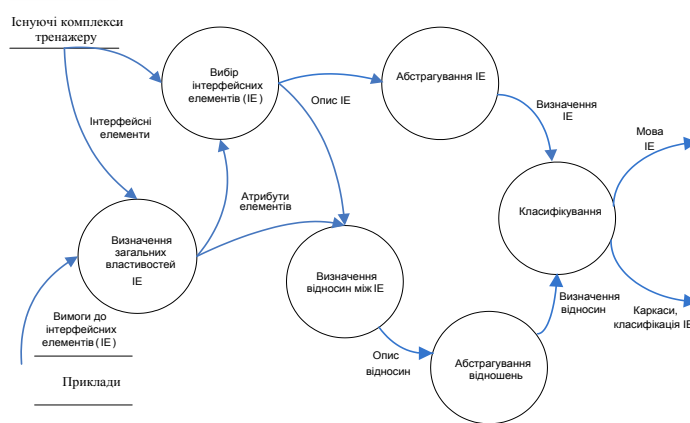


Рис. 4. Доменний аналіз інтерфейсних елементів

При побудові інтерфейсних елементів та настроювання інтерфейсу на різні види задач пропонується узагальнення доменної інформації на основі поняття ролі. В авіаційному прикладному домені роль визначається як сукупність функціональних обов'язків, властивих авіаційному фахівцю [10]. У процесі діяльності авіаційний фахівець отримує та забезпечує визначену інформацію через технічні системи. Сукупність видів інформації та способів її відображення, властивий одній авіаційній ролі будемо називати рольовим представленням. Таку інформацію можна представити як набір окремих інформаційних елементів, які групуються в логічно й ергономічно пов'язані групи. Останні будемо називати панелями представлень. Наприклад, до панелей представлень інструктора можна віднести навігаційну приладову панель, панель управління силовою установкою, навколишнє візуальне оточення кабіни літака.

Індикатори і органи управління пульта тренажера складають користувальницький інтерфейс інструктора, через який він отримує інформацію про політ та керує тренажером. Кожний індикатор відображає один параметр польоту. Множина індикаторів, доступних інструктору, складає реалізацію його рольового представлення і повинна визначатися підмножиною всіх параметрів польоту, й способами їх представлення, прийнятими в авіації, з урахуванням конкретного типу літака. Органи управління пульта визначаються потребами управління тренажером.

Набір і вид інтерфейсних елементів індивідуальні для тренажера літака кожного типу, однак суттєво схожі. Кожний інтерфейсний елемент  $ie = (it, pf)$  визначається двома складовими – видом інформації (параметром)  $it$  та формою її представлення  $pf$ . Рольове представлення  $rv$  можна описати як множину інтерфейсних елементів, що входять в нього  $RV = \{ie_i \mid ie_i \in RV\}$ , а роль  $R$ , у свою чергу – як множину панелей представлення  $R = \{RV \mid RV \in R\}$ . Вся множина інтерфейсних елементів домену  $IE = \{ie_i \mid ie_i \in IE, i = 1..d\}$  визначається як множина двійок  $ie = (it, pf)$  всіх його підмножин  $IE_r$ , визначених для ролей:  $R(IE) = \{IE_r \mid IE_r \subseteq IE\}$ . Інтерфейс для ролі інструктора синтезується шляхом вибору та настройки визначених інтерфейсних елементів:  $r_{instr} = RV_{instr} \subseteq \{ie_i = (it, pf) \mid i = 1..I\}$ .

#### Середовище програмування інтерфейсів пультів інструктора – результат доменного аналізу

Інтерфейсні елементи пульта інструктора (за призначенням) умовно поділяють на два класи: управляючі – для вводу інформації (перемикачі режимів, регулятори, кнопки вводу значень) та індикатори, призначені для її відображення (індикатори швидкості, висоти, зображення карт). У свою чергу індикатори набувають вигляду шкали-стрілки, цифрових табло та візирів. Множина інтерфейсних елементів складає інтерфейс.

Інтерфейсний елемент можна розглядати як композицію простих елементів: стрілок, шкал, діапазонів величин, маркування, кожний із яких характеризується певними властивостями і поведінкою (рис. 5).

Елементи індикатора представляються класами, атрибути яких характеризують певні властивості, наприклад, колір, радіус, діапазон величин, довжина стрілки.

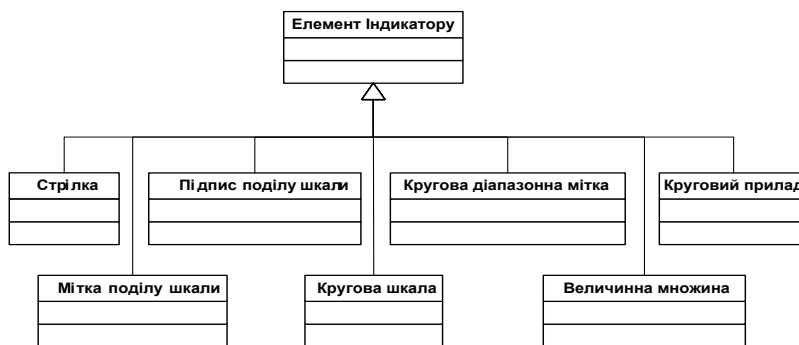


Рис. 5. Композиція елементів індикатора

Для побудови індикаторів різного виду був використаний шаблон проектування «компонувальник» (composite) [12]. Він об'єднує об'єкти в деревоподібну структуру для представлення ієрархії від окремого до цілого та дозволяє клієнтам звертатися до окремих об'єктів і до їх груп однаково. На рис. 6 а, б показана UML діаграма шаблону і, відповідно, окремий випадок індикаторного приладу. GaugeElement (індикатор) є абстрактним класом, що включає дочірні елементи Pointer (стрілка) та CircularScale (кругова шкала). Даний шаблон дає можливість здійснювати звернення до функціональності індикатора в цілому, а також до простих і складних елементів однаково. Так метод прорисовки Render викликає послідовно цей метод у всіх елементах композиції (Pointer, CircularScale), проходячи ієрархією класів.

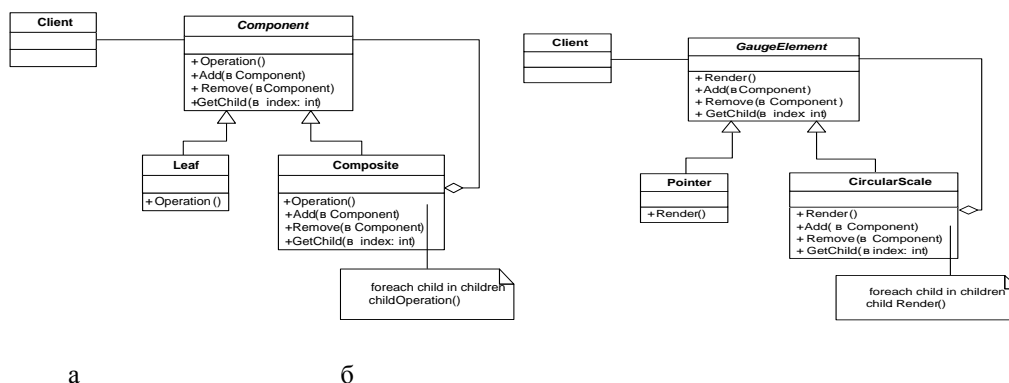


Рис. 6. UML діаграми для опису індикаторного приладу

У запропонованому підході індикатори будуються як композиції елементів. Наприклад, круговий прилад містить кругову шкалу, стрілку, мітку та підпис поділу шкали (рис. 7).

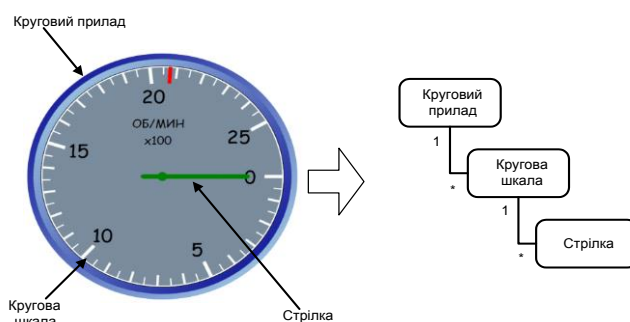


Рис. 7. Представлення інтерфейсного елемента кругового приладу

На основі класифікації і специфікування властивостей елементів у процесі доменного аналізу побудована ієрархія класів, на базі якої розроблено середовище. Шаблон «компонувальник», використовується як базова архітектурна концепція ієрархії (рис. 8).

Середовище класів реалізовано з використанням технології Windows Presentation Foundation. Для створення інтерфейсних елементів пульта застосовується мова розмітки XAML, що входить до складу Windows Presentation Foundation. Використання цих засобів дає можливість змінювати масштабованість індикаторів без втрати якості зображення, динамічно створювати нові індикатори та їх оформлення. Розробник може будувати індикатори відповідного типу, включаючи до композиції різноманітні елементи з бібліотеки.

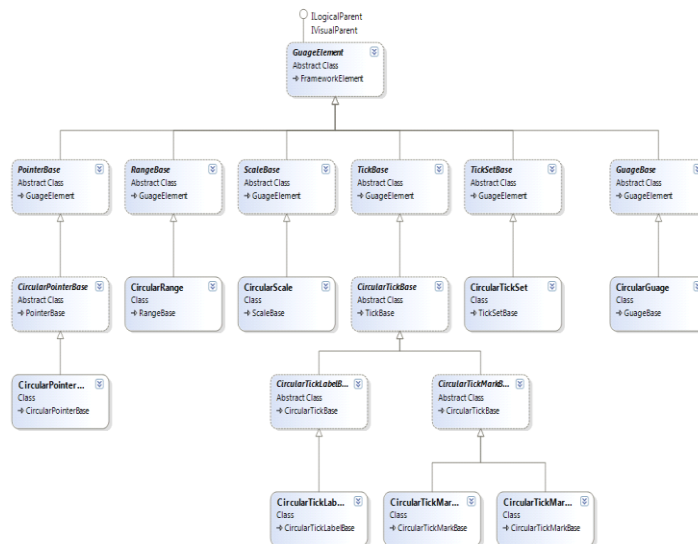


Рис. 8. Архітектура бібліотеки класів індикатора

Створюваний індикатор має деревоподібну структуру піделементів. Наприклад структура інтерфейсного елемента приладу «тахометр» (рис. 7), є наступною:

- круговий прилад (Circular Gauge);
- кругова шкала (Circular Scale);
- величинна множина (Tick Set);
- мітка поділу шкали (Tick Mark);
- підпис поділу шкали (Tick Label);
- стрілка (Pointer).

#### Відновлення працездатності програмного забезпечення систем контролю польотів авіації

На даний час авіаційні частини збройних сил України для об'єктивного контролю польотів користуються інформацією, яка отримується за допомогою різних типів окремих наземних систем дешифрування, не пов'язаних в єдину інформаційно-аналітичну систему [6]. Найбільшу частину працездатних наземних систем дешифрування польотної інформації складають застарілі системи аналогового типу «Луч-71М». Сучасні цифрові системи дешифрування представлені лише двома комплектами апаратури «Славутич» та одним комплектом – «Маяк-85». Система «Луч-84» призначена для обробки польотної інформації, накопиченої бортовими пристроями реєстрації параметрів типу МСРП і «Тестер». До складу системи «Луч-84» входить наступне: апаратура відтворення та вводу польотної інформації, накопиченої реєстраторами типу МСРП і «Тестер»; керуючий обчислювальний комплекс типу СМ 1420.08; алфавітно-цифровий друкувальний пристрій; пристрій графічної реєстрації даних; пристрій узгодження сполучень.

#### Формування вимог, щодо системи контролю

Різноманітність та застарілість чинних засобів реєстрації польотної інформації та наземних комплексів для її подальшої обробки, відсутність змоги відновлення й комплектування апаратної частини створюють значні перешкоди для регулярної та достовірної обробки.

Різноманітність програмного забезпечення для різних типів літальних апаратів не дають змоги об'єднання інформаційних потоків у єдину систему контролю польотів. Тому необхідне створення системи контролю польотів на нових засадах:

- використання уніфікованого програмно-апаратного забезпечення;
- створення баз даних для системного аналізу;
- обладнання літальних апаратів сучасними бортовими засобами реєстрації польотної інформації.

Розроблюваний програмно-апаратний комплекс має бути універсальним і давати змогу розшифровки та аналізу параметричної й мовної інформації, зареєстрованої різнотипними бортовими засобами реєстрації польотної інформації як у процесі експлуатації парку літальних апаратів, так і при розслідуванні льотних подій та інцидентів. Схема комплексу, яка повторно використовується, має давати можливість доповнювати його модулями, які забезпечують обробку інформації бортовими засобами реєстрації польотної інформації будь-яких типів. За допомогою доменного аналізу було отримано рішення у вигляді вимог та структури відповідного програмного забезпечення.

### Вимоги, щодо програмного забезпечення

Доменний аналіз показав, що для всіх типів літальних апаратів, оснащених бортовими засобами реєстрації польотної інформації, засобами комплексу вирішуються наступні задачі [6] :

- зчитування параметричної інформації з бортовими засобами реєстрації польотної інформації та розміщення її на жорсткому диску комп'ютера у вигляді сукупності файлів даних, які далі називатимуться копією параметричної інформації;
- зчитування мовної інформації з бортовими засобами реєстрації польотної інформації та розміщення її на жорсткому диску комп'ютера у вигляді файлів даних, які далі називатимуться копією мовної інформації;
- підготовка інформації за градувальними характеристиками датчиків бортовими засобами реєстрації польотної інформації та розміщення її на жорсткому диску комп'ютера у вигляді сукупності файлів даних, які далі називатимуться файлами градувальних характеристик;
- підготовка інформації щодо умов виконання польоту (маршрут, завантаження, центровка, метеоумови та ін.) і збереження її в одному з файлів копії параметричної інформації;
- збереження градувальної характеристики, чинної на момент обробки для даного літального апарата, в одному з файлів копії параметричної інформації;
- візуальний перегляд графіків змін параметрів польоту в часі на екрані відеомонітору з можливістю виділення окремих для кожної ділянки звернень при повторних переглядах;
- перегляд значень параметрів польоту у вигляді таблиць цифродруку;
- дослідження процесів польоту в інтерактивному режимі з допомогою запитів, що виконуються оператором на спеціальній мові алгоритмів експрес-аналізу;
- отримання копій графіків, таблиць цифродруку, бланків результатів досліджень на папері з допомогою принтера;
- прослуховування мовної інформації на аудіосистемі, підключеній до комп'ютера з візуалізацією графіка аудіосигналу та індикацією місця прослуховування на цьому графіку;
- прослуховування мовної інформації з різними швидкостями, з можливістю зацикловання ділянки, що зацікавила, та можливістю аналізу спектра аудіосигналу;
- аналіз траєкторії польоту ЛА за зображенням його у трьохвимірному просторі;
- швидкий пошук і запуск на обробку необхідної копії інформації з числа отриманих раніше за допомогою спеціального файлового процесора, орієнтованого на роботу з копіями параметричної та мовної інформації, але здатного також виконувати операції з файлами загального призначення;
- створення інформаційних баз даних з можливістю подальшого їх використання для системного аналізу на різних рівнях керування.

### Структура програмного забезпечення

Програмна частина комплексу, що виявлена в процесі доменного аналізу, має складатися зі взаємодіючих між собою програм та установочних (незмінних у процесі обробки одного польоту) файлів (рис. 9).

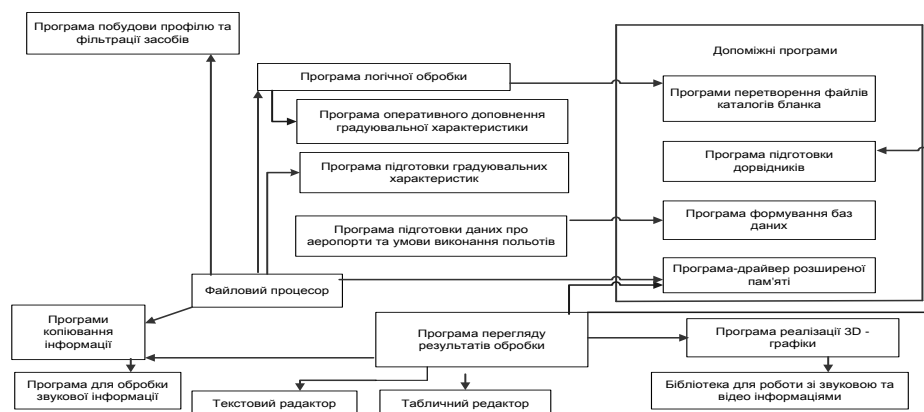


Рис. 9. Структура програмного забезпечення

До складу програмного забезпечення входить прикладне програмне забезпечення, спрямоване на аналіз польотної інформації та допоміжні програми. В якості останніх використовуються програми загального призначення (редактори текстів, редактори подвійних файлів, файлові процесори). Програми копіювання інформації призначені для зчитування інформації з відповідних наземних засобів відтворення та створення копій параметричної інформації у вигляді сукупності файлів прямої копії. Програма побудови профілю польоту та фільтрації збоїв, що призначена для виділення з копії інформації про стандартизовані ділянки польоту (зліт, посадка та ін.), для виключення збійних значень параметрів польоту на цих ділянках та для визначення моментів ряду характерних точок польоту (запуски двигунів, доторкування при посадці та ін.).

**Реалізація та застосування засобів зворотної інженерії**

Реалізація засобів здійснювалася в Національному авіаційному університеті в процесі відновлення працездатності авіаційного тренажера ТЛ-410М для реінженерії його програмного забезпечення [5, 7].

Засіб відновлення інформації про програмне забезпечення, згідно з загальною схемою (рис. 3), було розроблено так, що він містить графічний інтерфейс і ядро, яке, в свою чергу, містить два компоненти – екстрактор та абстрактор. Успадкований початковий код в автокоді SYPS транслюється в текст на проміжній мові DOT [6], який графічним модулем Graphviz [13] перетворюється на графічне представлення алгоритму початкового коду (блок-схему) у вигляді \*.gif файлу. Інтерфейс відображає блок-схему для користувача (рис. 10).

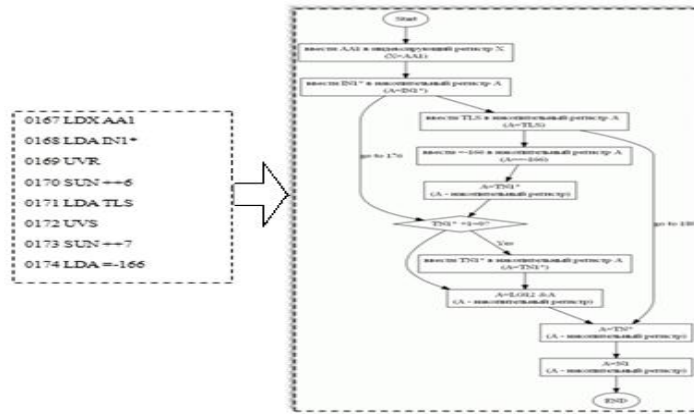


Рис. 10. Приклад результатів роботи засобу

Засоби відновлення інформації про технічне обладнання – це інструменти зворотної інженерії, які призначені для дослідження технічного обладнання з метою узгодження параметрів, значення яких виробляє відновлене програмне забезпечення і характеристик відповідного технічного обладнання комплексу. Засоби забезпечують дослідження різноманітного обладнання комплексу та містять компоненти, які можна поділити на дві групи. До першої входять засоби досліджень індикаторів, до другої – датчиків. Нині в якості цих засобів використовують відповідні технічні прилади, а тому побудова характеристик обладнання здійснюється вручну.

Пропонуються програмні засоби, які автоматизують не тільки дослідження, але й обробку результатів (рис. 11). Для реалізації дослідження пропонується використовувати спеціальний набір інструментів, що складається з вимірника сигналів і генератора тестових сигналів.

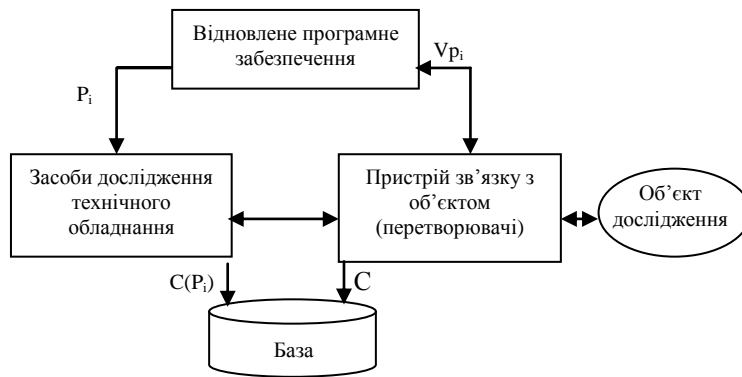


Рис. 11. Схема застосувань засобів дослідження технічного обладнання

Вимірник сигналів, призначений для зняття характеристик органів управління і датчиків, забезпечує набуття цифрових значень сигналів (D), нормованих значень параметрів (P), відповідних значень напруги (U), а також візуалізацію значень і графіків (рис. 12).

Селектор параметрів забезпечує вибір перетворювача і каналу, через який в обчислювальну систему поступають значення сигналу органу управління. Цифрові значення відліків (D) знімаються з перетворювачів, а нормовані значення параметра обчислюються наступним чином:  $P_i = f_p(D, C_{pi})$ , де  $C_{pi}$  – конфігурація параметрів. Цифрові значення відліків (D) перетворюються на значення напруги (U), приведені на вхід перетворювача з використанням даних конфігурації відповідного пристрою вводу-виводу (C) –  $U = f_u(D, C)$ .

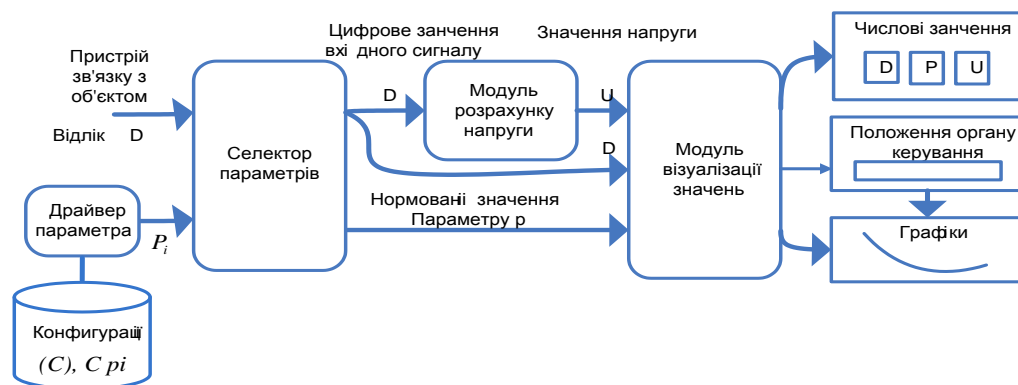


Рис. 12. Структура програмного забезпечення вимірювача сигналів

Модуль візуалізації значень забезпечує виведення значень сигналу, що знімаються з органу управління у вигляді цифрових значень відліків (D), нормованих значень параметра (P) та значень напруги (U). Інструмент будує характеристики органу управління шляхом відслідковування послідовної дискретної зміни його положення, а також графічно відображає зміну значень D, P, U в часі. Генератор тестових сигналів виробляє та подає тестові сигнали на індикаторні прилади й виконавчі пристрої (рис. 13). Передбачена можливість задавати для управління виходом пристрою будь-який з трьох типів значень – D, P, U. Значення, що вводяться через селектор, подаються безпосередньо в драйвер пристрою або параметрів, відповідно. Значення напруги подаються через модуль перерахунку напруги.

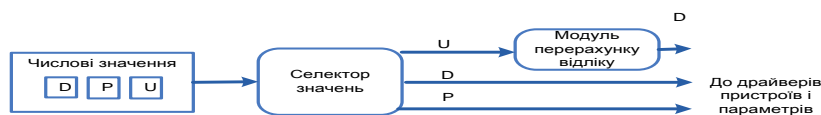


Рис. 13. Структура програмного забезпечення генератора тестових сигналів

## Висновки

На прикладі авіаційного домену наведено досвід реінженерії програмного забезпечення авіаційних тренажерів і систем контролю польотної інформації. Застосування доменного аналізу та зворотної інженерії хоча й потребують власних методів і засобів, але дозволяють побудувати такі рішення, які можна повторно використовувати в домені при реалізації програмного забезпечення авіаційної техніки, придатність якої намагаються продовжити або підтримати.

1. Пащенко С.В., Хільченко М.Ф. Система підтримки справності та надійності авіаційної техніки при її експлуатації за технічним станом. Матер. наук.-практ. конф. «Актуальні проблеми розвитку авіаційної техніки». – Київ. – 2009. – С. 87
2. Самуєв В.В. Діяльність командування повітряних сил збройних сил України щодо підтримання справності та боєздатності авіаційної техніки та основні пріоритетні напрямки наукових досліджень. Матер. наук.-практ. конф. «Актуальні проблеми розвитку авіаційної техніки». – Київ. - 2009. – С. 99.
3. Старосила М.І. Актуальні аспекти продовження експлуатації літальних апаратів за межами установлених строків служби. Матер. наук.-практ. конф. «Актуальні проблеми розвитку авіаційної техніки». – Київ. – 2009. – С. 104.
4. Chikofsky E.J., Gross J.H. Reverse Engineering and Design Recovery: Taxonomy // IEEE Software. – 1990. – P. 13–17.
5. Сидоров Н.А., Недоводеев В.Т., Хоменко В.А.и др. Реинженерия наследуемого программного обеспечения информационно-моделирующих тренажерных комплексов // Управляющие системы и машины. – № 4. – 2008. – С. 68–74.
6. Rugaber S., Stivewalt K. Mode-Driven Reverse Engineering. – IEEE Software. – 2004. – P. 45–52.
7. Луцький М.Г., Сидоров М.О., Зятдінов Ю.К. Підтримка придатності інформаційно-програмного забезпечення авіаційної техніки // Матер. Наук.-практ. конф. «Актуальні проблеми розвитку авіаційної техніки». – Київ. – 2009. – С. 75.
8. Prieto-Diaz R. Domain Analysis: An Introduction.– Software engineering Notes. – 1990. – V. 15. – № 2. – P. 47–54.
9. Луцький М.Г., Хоменко В.А., Сидоров Є.М. Засоби зворотної інженерії програмного забезпечення авіаційних інформаційно-модельюючих комплексів // Матеріали ІХ Міжнар. наук.-техн. конф. «АВІА – 2009». – 21–23 вересня 2009, Київ.– НАУ.
10. Программы подготовки летного состава гражданской авиации (ППЛС ГА – 92).
11. Design of a flight simulator software architecture. Goran Anchen, Han Wallenberg.– School of Mathematical and Systems Engineering, Vaxjo University.– 2002.– 91 p.
12. Ларман К. Применение UML и шаблонов проектирования. – М.: «Вильямс», 2002. – 624 с.
13. E.R. Ganser, S. C. North. An open graph visualization system and its applications to software engineering//Software Practice and Experiments, V. 1.– 1999. – P 1–5.