УДК 004.5.

РОЗРОБКА МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ ТА ПРОГРАМНИХ МОДУЛІВ ДЛЯ КІС ТПВ

Д.т.н. І.Ш. Невлюдов, к.т.н. В.В. Євсєєв, А.І. Демська, Харківський національний університет радіоелектроніки

У роботі запропонована модель життєвого циклу «Jump» для розробки програмних продуктів і програмних модулів для комп'ютерно-інтегрованих систем технологічної підготовки виробництва. Наведено формалізація моделі у вигляді кортежу параметрів, описані всі елементи і їх призначення, на базі формалізованої моделі розроблений метод синтезу компонентів і структури основних об'єктів, елементів опису модельованого програмного продукту або програмних модулів для скорочення часу проектування і розробки на ранній стадії технічного завдання.

В работе предложена модель жизненного цикла «Jump» для разработки программных продуктов и программных модулей для компьютерно-интегрированных систем технологической подготовки производства. Приведена формализация модели в виде кортежа параметров, описаны все элементы и их назначение, на базе формализованной модели разработан метод синтеза компонентов и структуры основных объектов, элементов описания моделируемого программного продукта или программных модулей для сокращения времени проектирования и разработки на ранней стадии технического задания.

The paper proposes a model of the life cycle "Jump" for the development of software products and software modules for computer-integrated systems of technological preparation of production. The formalization of the model in the form of a tuple of parameters, all the elements and their purpose are described; on the basis of the formalized model, the method of component synthesis and the structure of the main objects, the elements of the description of the simulated software product or program modules for reducing the design and development time at an early stage of the technical task is developed on the basis of the formalized model.

Ключові слова: *автоматизація, програмний продукт, програмні модулі, життєвий цикл*

Вступ

Основою розвитку сучасного приладобудування у світі є комп'ютеризація та інтеграція всіх виробничих процесів та управління виробництвом від початку розробки до поставки готової продукції споживачеві.

Інтеграція у виробничих системах або комплексах (в широкому сенсі, як це тепер розуміється в рамках концепції міжнародних стандартів ISO серії 9000) незалежно від категорії і виду виробничої діяльності та галузі народного господарства, а також рівня і масштабу інтеграції (починаючи з нижчого рівня, інтеграції операцій на одному робочому місці і закінчуючи інтеграцією на найвищому, міжнародному рівні).

Якщо спиратися на ідеологію, відповідну зазначеним міжнародним стандартам, то слід в першу чергу говорити про інтеграцію з метою вдосконалення діяльності щодо забезпечення всіх етапів життєвого циклу (ЖЦ) (англ, life-cycle), на чому ґрунтується сучасна теорія управління якістю [1].

В даний час центром ваги в інтеграції вважається використання уніфікованих комп'ютерних технологій та програмного забезпечення різноманітної документації (проектної, технологічної, робочої (безпосередньо відноситься до виготовлення), експлуатаційної та ін.) та відповідного програмного забезпечення.

Основною метою даного дослідження є створення методологічно обґрунтованої технології автоматизованого проектування програмних продуктів (ПП) та модулів (ПМ) для комп'ютерно-інтегрованих систем технологічної підготовки виробництва (КІС ТПВ).

Механізм, що забезпечує автоматизацію, на базі реінжинірингу структури мов високого рівня програмування і середовищ розробки для розробки нової моделі ЖЦ, який забезпечує скорочення часу проектування і реалізації, досягається на стадії складання технічного завдання. Це привело до необхідності перегляду існуючих підходів, методологій і технологій проектування та розробки програмних продуктів для КІС ТПВ, а також потребу створення нових універсальних моделей і методів забезпечення автоматизації проектування, які дозволять уникнути витратної політики виконання зайвих і циклічних робіт при проектуванні програмних продуктів і модулів в рамках розв'язання задач для КІС ТПВ.

Розробка моделі ЖЦ ПП «Jump»

Ґрунтуючись на проведеному аналізі можна зробити висновок, що існуючі аспекти та рекомендації, які представлені в сучасних міжнародних стандартах, не дозволяють використовувати їх в рамках комплексного підходу для вирішення завдання розробки єдиної методології в області автоматизації проектування ПП та ПМ для КІС ТПВ.

Всі існуючі стандарти мають часткові пропозиції і рекомендації щодо систематизації розробки програмних продуктів загального призначення.

Виходячи з цього, для реалізації єдиної методології автоматизації проектування програмних продуктів для КІС ТПВ, в рамках даних досліджень, пропонується нова модель ЖЦ «Jump», яка представлена на рисунку 1.

© І.Ш. Невлюдов, В.В. Євсєєв, А.І. Демська

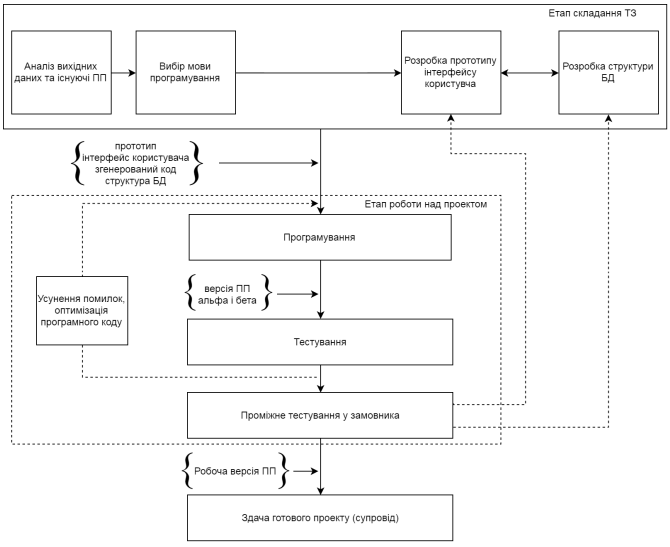

Рис. 1. Модель ЖЦ ПП «Jump»

Перевага даної моделі полягає в тому, що вона складається з трьох основних етапів:

- етап складання технічного завдання (ТЗ);
- етап роботи над проектом;
- етап здачі і супроводу програмного продукту.

Пропонована модель дозволяє в собі об'єднати існуючі концепції і підходи, синтезувати їх, з точки зору застосування в нову методологію автоматизації проектування ПП і ПМ для КІС ТПВ.

Розглянемо детально кожен етап ЖЦ ПП «Jump» як сукупність підсистем, що знаходяться у відносинах і зв'язках між собою і утворюють цілісну послідовність для вирішення завдання комплексної методології автоматизації проектування програмних продуктів та модулів для КІС ТПВ [2].

Запропонована модель ЖЦ відповідає наступним принципам: єдності та ієрархії, має відкриту ступінь взаємодії із зовнішнім середовищем і володіє властивостями самоорганізації при зміні специфіки завдань і вимог до мов високого рівня програмування і середовищ розробки, в залежності від вимог, що висуваються до неї.

Для опису всіх зв'язків в підсистемах і всередині системи запропоновано використовувати метод структуризації (МАІС - методи, спрямовані на активізацію інтуїції і досвіду фахівців) на базі теоретико-множиних уявлень і інформаційному (метод оцінки ступеня цілевідповідності аналізованих складових досліджуваних систем за різнорідними якісними критеріями, що дозволяють отримувати узагальнену оцінку в багатокритеріальних задачах з різнорідними критеріями) підході формалізації процесів, зв'язків, які протікають всередині моделі і кожного етапу (МФПС - методи формалізованого представлення систем). Даний підхід дозволить вирішити такі важливі завдання:

- забезпечити представлення моделі ЖЦ «Jump» на такому рівні абстракції, на якому вона стає виразною і інтуїтивно зрозумілою для аналітиків, проектувальників КІС ТПВ, Software Product Manager, розробників (програмістів) і кінцевим користувачам;
- дозволяє проектувальникам КІС ТПВ уточнювати її структуру і поведінкову організацію незалежно від середовища розробки і призначення розроблюємих ПП або модулів в рамках КІС ТПВ.

На підставі аналізу вимог, що висуваються для вирішення поставлених завдань, які вимагають необхідність знаходження нових рішень сформульованої наукової проблеми для досягнення мети розробки нової методології в області автоматизації проектування ПП і ПМ для КІС ТПВ, були сформульовані вимоги, що висуваються до моделі ЖЦ «Jump»:

- розроблена модель повинна володіти необхідною і достатньою спільністю, щоб її засоби давали можливість забезпечити прозоре, адекватне, комплексне представлення даних і зв'язків для всіх учасників проекту з виконанням умов єдиної структури і цілісності;
- розбіжність пропонованої моделі ЖЦ «Jump» і об'єктно-орієнтованих підходів на основі якої реалізується ПП і ПМ, пристосована до динамічних змін предметної області повинна бути мінімальна, а отже склад і структура набору формальних об'єктів створюваної в моделі ЖЦ і об'єктно-орієнтованої повинні бути максимально близькими, щоб при відображенні концептуальної моделі ПП або ПМ для КІС ТПВ можна було скористатися відомими правилами перетворення структур, побудови, ієрархії без втрат цілісності;
- можливість комплексного використання моделі ЖЦ як на етапі проектування ПП і ПМ для КІС ТПВ, так і на стадії розробки та функціонування, як основи для інтерфейсів користувача.

Для задовільнення сформульованих вище вимог виділимо чотири основні етапи відповідно до загального підходу, викладеного в [3]:

- виявлення деяких латентно-семантичних понять (концепцій) (LSA), які будуть використовуватися для опису моделі;
- визначення набору формальних об'єктів, які можуть використовуватися для представлення вибраних понять;
- визначення формальних правил підтримки цілісності даних в моделі ЖЦ;
- визначення формальних операторів взаємодії моделі ЖЦ «Jump» з «реальним світом», між етапами як підсистемами даної моделі, а також взаємодія всередині кожної підсистеми.

Інформація про розглянуту предметну область («реальний світ») представляється через сприйняття розробника або аналітика, і складається з безлічі взаємозалежних і взаємопов'язаних факторів, що орієнтує на розгляд моделі ЖЦ з позицій закономірності системного цілого і взаємодії його складових сутностей (етапів). Ґрунтуючись на концепції багаторівневої ієрархії суті моделі ЖЦ, процеси і явища доречно розглянути як безліч дрібніших ознак (підмножин) опису, при цьому етапи логічно розглядати як елементи більш високих класів узагальнень. Виходячи з запропонованої моделі ЖЦ «Jump», вона повинна забезпечувати адекватне уявлення і правильне сприйняття, отже, доцільно використовувати принцип, з яким таке уявлення в першу

чергу будує для себе аналітик або розробник природною мовою.

Ґрунтуючись на визначенні, що всі процеси «реального світу» протікають в просторі і в часі, то модель ЖЦ являє собою динамічну систему, яка складається з певних послідовностей станів, де в кожен момент часу визначається його множиною кінцевих станів. Кожна множина кінцевих станів описується елементарними факторами природною мовою в термінах об'єкта, зв'язків і приладдя об'єкту, значення параметрів і його властивостей під час виконання тієї чи іншої події. Внаслідок цього опис моделі ЖЦ «Jump» можна представити з точки доцільності в термінах предметної області, розширивши і синтезувавши їх набір. В рамках даного дослідження і в контексті розробленої методології і моделі ЖЦ «Jump» запропоновані наступні поняття:

- *об'єкт* пов'язується з різними частинами моделюємого програмного продукту або модулями для КІС ТПВ, властивість (опис характеристик) об'єкта дозволяє співвіднести безліч об'єктів (підкласів) і їх властивостей з одним загальним класом;
- *подія* - реакція відбувається на об'єкті (певного підкласу), кожна подія описується його властивостями і визначається моментом часу його настання або виклику;
- *значення* характеристик об'єкта і події необхідні ознаки (параметри, що відносяться певним класом і їх підкласом) уявлення об'єкта (опис його візуалізації) або реакції на події у вигляді певної необхідної дії.

Ці поняття є неформальними концепціями (метапоняттями) моделі як невід'ємний елемент відповідно до підходу за К. Дейтом [3]. При цьому варто зауважити, що основоположною, в розробленій моделі ЖЦ, умовою є існування «об'єкта» (форми) і подій, які можуть належати як об'єкту як класу, так і підкласам, які входять в них. Виникнення об'єкта обумовлено інтересом суб'єкта і зникає коли втрачається інтерес до об'єкта суб'єктом. В [3] визначено єдину властивість існування «об'єкт-час» його виникнення, зникнення і зміни, пов'язане з подіями, що відбуваються з ним.

У розробленій моделі ЖЦ «Jump» знайшли застосування певні підходи, рішення і методології, які вже апробовані: методологія RUP (Rational Unified Process), розроблена компанією Rational Software; методологія Microsoft Solution Framework, запропонована компанією Microsoft; методологія Scrum (управління проектами), яка заснована на принципах Time-менеджменту; гнучка методологія Agile (розробки з швидкого проектування і розробки ПП і ПМ), де головним завданням ставиться працюючий продукт, а не його документація; методологія візуального об'єктно-орієнтованого програмування (абстрагування, поліморфізм, інкапсуляція), семантично мережевих, онтологічної, інфологічної, логіці предикатів з розширеною підтримкою в часі.

Відштовхуючись від того, що парадигма методологій візуального об'єктно-орієнтованого програмування, концептуальної та логічної моделі близькі, це дозволяє спростити перехід від концептуальної моделі до логічної. Також на реалізацію моделі ЖЦ вплинули семантичні моделі, які дають можливість опису і представлення складних ситуацій, концепція ієрархій типів, які дозволяють обґрунтувати і систематизувати уявлення класів, метакласів, концепція ролі понять і часу, що дозволяє реалізувати висунуті вище вимоги до моделі ЖЦ «Jump».

Формалізація моделі ЖЦ «Jump»

У відповідності до певних вимог і понять, модель ЖЦ «Jump» ($\aleph$) можна представити у вигляді наступного кортежу на базі четвертого виду представлення системи на основі класичної теорії системного аналізу:

$$\aleph = \left\langle P(\Im, \Re, H), K, L \right\rangle, \qquad (1)$$

де $P(\Im, \Re, H)$ - сукупність правил і структурування даних про об'єкти моделювання; P - об'єкт моделювання (ПП або МП); $\Im$ - безліч базових понять (концепцій уявлення) об'єкта моделювання; $\Re$ - безліч відносин між базовими поняттями моделі; H - безліч функцій описів і трактування базових понять і відносин; K - безліч вимог обмеження цілісності; L - мова представлення моделювання даних.

Для подальшого опису сукупності правил і структурування даних про P, необхідно розробити словник опису і трактування базових понять (H), який складено для множини базових понять $\Im$.

Уявімо множину $\Im$ як набір базових понять в такому вигляді:

$$\Im = \{Form, ParameterForm, ValueForm, EventForm, LinguisticVariable, ElementForm, ParameterElement, ValueElement, EventElement, ContainerSolutions\} \qquad (2)$$

Ці поняття є основними елементами уявлення деякого формального об'єкта моделювання. При розробці опису модельованого ПП або ПМ кожне з (2) буде містити конкретні імена (назви), які будуть запозичуватись зі словників і визначень того чи іншого середовища розробки, або неформальним шляхом узгодження базових понять між розробником і користувачем автоматизованої системи проектування ПП і ПМ для КІС ТПВ. Внаслідок цього, вони є множиною відносин між однойменними множинами, співвідносними з даними поняттями і їх елементами як формальними об'єктами, які дають можливість представляти властивість елементів модельованих ПП і ПМ, їх взаємодію і зв'язок, за умов, що модель, що розроблюється, має кінцеву множину таких зв'язків і взаємодій. Далі наведемо опис основних базових понять з виразу (2), визначимо неформальне визначення та призначення в рамках формалізації об'єкту моделювання на базі запропонованої моделі ЖЦ «Jump».

Розробка методу синтезу компонентів і елементів структури ТЗ на ранній стадій моделі ЖЦ «Jump»

Для автоматизації проектування програмних продуктів на базі ТЗ необхідно розробити структуру листа ТЗ, яка забезпечить повноту опису вхідних даних і

логічний взаємозв'язок між ними і властивостями основних параметрів, які властиві для елементів мов високого рівня програмування і використовують інтерфейс користувача.

Визначимо P - як порядкову множину, призначену для користувача форм ($Form_i$), яка виконує умову $\exists Form_i \in P$ при $i = 1,...,n$ де $n \le 40$, максимальна кількість яких не залежних від головної форми вікон інтерфейсу, з якого складається проектований програмний продукт і взаємодіє з головною формою ($Form_i$). Основні елементи $Form_i$ описані в стандартах [4].

Виходячи з цього визначимо об'єкт $Form_i$ де $i = 1,\ldots,n$, як безліч:

$$Form_i \in \{Form_i \text{ Pr}opertied\ and\ evens,\ Components\ \ on\ Form_i structure\} \quad (3)$$

при умові що $Form_i$ Pr*opertied and events* і *Components on* $Form_i$ *structure* $\neq \emptyset$.

Теорема 1. Про існування $Form_i$ тільки за умов $Form_i$ Pr*opertied and events* і *Components on* $Form_i$ *structure* $\neq \emptyset$.

Доказ: Припустимо *Components on* $Form_i$ *structure* $\neq \emptyset$, то отже на $Form_i$, як на робочому вікні інтерфейсу, не будуть присутні елементи візуалізації роботи з даними (елементи Button, Grid, і т.д.), отже інтерфейс розроблюваного програмного продукту не має функціонального призначення і його розробка не доцільна і це логічно. Отже виходячи з цього *Components on* $Form_i$ *structure* $\neq \emptyset$ і це тотожність ліквідна для $Form_i$ Pr*opertied and evens* і дозволить уникнути парадоксу. Теорема доведена.

Визначимо $Form_i$ Pr*opertied and evens* як множину параметрів *Main propertied* і значень Pr*operties parameters* притаманних кожній $Form_i$, як невід'ємну частину $Form_i$ Pr*opertied and evens*. Уточнимо, що кожному параметру множини *Main propertied* належить хоч одна підмножина значень Pr*operties parameters*, але при цьому деякі значення підмножини можуть бути Pr*operties parameters* $= 0$. Внаслідок цього можна записати

$$\exists \text{Pr}operties\ parameters_i \in \text{Pr}operties\ parameters: \quad (4)$$
$$: \text{Pr}operties\ parameters_i \neq 0$$

Наведемо підмножини *Main propertied* і Pr*operties parameters* як упорядкований жорстко фіксований порядок опцій $mp_i \in Main\ propertied$, де mp_i - параметр як невід'ємна частина підмножини *Main propertied*$\in Form_i$ Pr*operties and evens* і відповідно множини значення $pp_i \in \text{Pr}operties\ parameters$, які в свою чергу задовольняють умові, що Pr*operties parameters* $\in$ $Form_i$ Pr*opertied and evens*. Введемо безліч Z з елементами $(z_1, z_2,..., z_k)$ як множину можливих варіацій значень (розмір $Form_i$ в ріх, розташування «Top», «Left», резервування слів «alNone - немає вирівнювання», і логічних «True» «False» і т.д.) які залежать від синтаксису опису $Form_i$ для певної мови програмування і середовища розробки. Виходячи з цього визначимо $\exists z_i \in Z : pp_i \neq 0$. В окремих випадках можливе при умові що $z_i \in Z \subseteq pp_i \in$ Pr*operties parameters*$\neq \emptyset$. Також невід'ємною частиною множини є підмножина *Main events*, яка представляє набір параметрів $Form_i$ Pr*opertied and evens*, таких як «Crate», «Close», «Onclick» і т.д., для опису їх введемо me_i де $(me_1, me_2,..., me_n) \in Main\ events$ як деякі допустимі дії над множиною $Form_i$. Безліч параметрів обмежується правилами, які задані в тому чи іншому середовищі розробки. Доведе існування me_i як елемент безлічі *Main events* з наступним правилом існування:

$$Main\ events = \{me_i \in Main\ events : Form_i \text{ Pr}opertied\ and\ evens \neq \emptyset\} \quad (5)$$

Кожному елементу me_i властива безліч *Events on action whith Form*, яка складається з нескінченної множини елементів ea_i, як можливий набір «шаблонів», у вигляді програмного коду, який містить в собі всі допустимі процедури / функції, які можуть виконуватися елементом me_i для кожної мови програмування. Наведемо це як:

$$Events\ on\ action\ whith\ Form \in \{ea_1, ea_2,..., ea_i\} \quad (6)$$

На базі висловлених припущень представимо множину $Form_i$ Pr*opertied and evens* як:

$$\{(\exists pp_i \cap mp_i \in Main\ properties{:}Main\ properties \neq \emptyset)\ \&\ \&(\exists ea_i \cap me_i \in Main\ elements \neq \emptyset)\} \subseteq Form_i \text{ Pr}opertied\ and\ evens \quad (7)$$

Вираз (7) не достатній і не повний для вирішення завдання поставленого в даному дослідження, так як не враховує візуальні компоненти і компоненти роботи і відображення даних. Внаслідок цього введемо множину *Components on* $Form_i$ *structure*$\in Form_i$ при умові що як множина (*Components on* $Form_i$ *structure/*$Form_i$ Pr*opertied and evens*$\in Form_i$) елементів опису візуальних компонентів інтерфейсу необхідна і достатня для реалізації управління з даними $Components\ description_i \in Components\ on\ Form_i structure$, де $i = 1,\ldots,n$ - кількість візуальних компонентів властивих на $Form_i$ і виконують певні функцій. У свою чергу для опису візуальних компонентів введемо підмножини Pr*eporties components*$_i$ і *Component events on action*$_i$ як невід'ємні частини множини *Components description*$_i$, де $i = 1,2,...,n$ кількість різноманітностей візуальних форм, за допомогою яких можна працювати з інформацією.

Внаслідок цього будь-який візуальний елемент можна описати в такому вигляді:

$$Components\ description_i \in \in \{\Pr eporties\ components_i, Component\ events\ on\ action_i\} \quad (8)$$

Для опису властивостей множин $\Pr eporties\ components_i$ і $Component\ events\ on\ action_i$ введемо параметр $pc_i \in \Pr eporties\ components_i$ де $i = 1,...,n$ кількість параметрів, які описують властивості об'єкта pc_i, може приймати значення безлічі Z які описані вище. На відміну від $Main\ properties_i$, що єдиний для $Form_i$, то кількість $Components\ description_i \in Components\ on\ Form_i structure$ має бути $n \geq 1$ де $n = 1,...,m$, в іншому випадку при не виконанні даної умови спрацьовує теорема 1.

Висновки

Розроблений метод синтезу компонентів і елементів модельованого ПП або ПМ на базі формалізації моделі ЖЦ «Jump» для проектування КІС ТПВ. Дане рішення дозволяє зробити детальну декомпозицію всіх елементів ПП або ПМ, врахувати і виділити як необхідну структуру побудови і зв'язків кожного елемента всередині системи так і обґрунтувати їх необхідність існування, задати всі необхідні параметри і події.

У майбутньому, на базі запропонованого математичного опису та подання буде розроблена методологія проектування і розробки ПП і ПМ для КІС ТПВ і реалізована у вигляді автоматизованої системи проектування і розробки САх систем.

Основною метою автоматизованої системи проектування і розробки ПП для КІС ТПВ, що розробляється, є зменшення трудомісткості і ризиків, за рахунок автоматизації і часткової реалізації інтерфейсу користувача та програмного коду на ранній стадії складання ТЗ.

СПИСОК ЛІТЕРАТУРИ:

1. Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В., Мілютіна С.С. Основи САПР: технічна підготовка виробництва: Навчальний посібник/ І.Ш. Невлюдов. - Київ: НАУ, 2014. – 360 с.

2. Черніков Ю.Г. Системний аналіз і дослідження операцій. - Москва, МДГУ, 2006.-370 с.). (ISBN 5-7418-0424-1)

3. К. Дейт Руководство по реляционной СУБД DB2/Пер. С анл. И предисл. М.Р. Когаловского. М.: Финансы и статистика, 1988. – 320 с. (ISBN 5-279-00036-9)

4. Волченков Е., Стандартизація пользовательского интерфейса Электронный ресурс. / Евгений Волченков // Журнал «Открытые системы» № 4 / 2002 — Режим доступа: http:// www.osp.ru/ os/ 2002/ 04/181312/

УДК 621.37/39.019.3

ДИАГНОСТИРОВАНИЕ АНАЛОГОВЫХ МОДУЛЕЙ РАДИОЭЛЕКТРОННЫХ СИСТЕМ

К.т.н. Б.А. Шостак[1], к.т.н. В.А. Письменецкий[2], С.И. Теслюк[2]

1. Национальный технический университет «Харьковский политехнический институт», г.Харьков

2.Харьковский национальный университет радиоэлектроники

В статье приведены результаты исследований по диагностике аналоговых модулей радиоэлектронных систем посредством применения метода анализа единичных откликов.

В статті приведено опис і основні характеристики методу примусового діагностування аналогових модулів радіоелектронних систем.

The article is developed new method of the diagnostics, designed for the solving of the problems, intended for optimization of searching for failure analog element in module of any complication and functional types.

***Ключевые слова**: процесс диагностирования, аналоговый модуль, априорная неопределенность.*

Введение

Для эффективного решения задачи обнаружения возникших в процессе эксплуатации дефектов, как правило, необходимо уже при проектировании обеспечить диагностируемость радиоэлектронных систем (РЭС). Но при этом возникают определенные трудности. Эти трудности обусловлены тем, что при моделировании сложных объектов действует фундаментальное положение кибернетики: при сложности объекта выше некоторого уровня его адекватная (полная модель) не может быть более простой. Поэтому фактически ни одна электронная система, содержащая модули, выполняющие преобразование аналоговых сигналов, не имеет исчерпывающего математического описания.

Наибольшее применение при решении задачи диагностики нашли следующие методы и технологии:

- DFT (Design for Test) или технология проектирования контролепригодных схем – технология, которая упрощает разработку и проведение производственных испытаний, а также обеспечивает диагностируемость электронного оборудования.

Большинство вариантов DFT, применяемых сегодня, основаны на принципе структурного испытания. Структурное испытание не создает прямого воздействия для определения, все ли выходные функции схемы соответствуют норме. Вместо этого осуществляется проверка, что схема правильно собрана из некоторых низкоуровневых составных частей в соответствии со структурной таблицей соединений. При этом должно

© Б.А. Шостак, В.А. Письменецкий, С.И. Теслюк