

**Висновки**

Можливості, які надають сучасні комп'ютери розробникам радіоапаратури, дозволяють говорити не тільки про аналіз, але й про синтез пристроїв НВЧ на електродинамічному рівні. Розвиток та реалізація методів аналізу нелінійних пристроїв призводять до виключення всякого роду спрощень. Розрахунок, моделювання та вимірювання пристроїв НВЧ об'єднуються в єдиний процес і потребують високої кваліфікації спеціаліста.

Сучасний технічний прогрес неухильно веде до ускладнення систем, приладів, комплексів. Тому й така традиційна форма навчання на радіотехнічних факультетах, як лабораторна робота змінює свою форму – від зняття показників приладів до моделювання систем на комп'ютері. Студенти отримують знання, аналогічні знанням отриманим при використанні реальних приладів, використовуючи комп'ютерні моделі.

Використання *Awr Design Environment* в процесі підготовки висококваліфікованих інженерів дає можливість значно підвищити ефективність освоєння НВЧ техніки студентами, а також ознайомлює майбутніх фахівців з потужним інструментом проектування НВЧ систем.

**Література**

1. Chen L. F. *Microwave Electronics: Measurement and Materials Characterization*/ [Chen L. F., Ong C. K., Neo C. P., Varadan V. V., Varadan V. K.] United Kingdom, John Wiley & Sons Ltd., 2004. – 537ps.
2. Thury J. *Microwaves: Industrial, Scientific and Medical Applications*/ Thury J. – Boston – London, Artech House, 1990. – 669ps.
3. Разевиг В. Д. Проектирование СВЧ устройств с помощью Microwave Office / Разевиг В. Д., Потапов Ю. В., Курушин А. А. – М.: СОЛОН-Пресс, 2003. – 496 с.
4. Фриск В. В. Основы теории цепей. Использование пакета Microwave Office для моделирования электрических цепей на персональном компьютере / Фриск В. В. – М.: СОЛОН-Пресс, 2004. – 160 с.

Надійшла 6.9.2010 р.

УДК 004.4

**В.І. МЕЖУСЬ**

Одеський Національний політехнічний університет

## **ПРЕДМЕТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ТА ОРГАНІЗАЦІЯ ПРОЦЕСІВ СИСТЕМНОЇ ІНЖЕНЕРІЇ**

*У статті розглядаються принципи створення й застосування предметно-орієнтованих моделей різних процесів системної інженерії. Запропонований підхід дозволяє моделювати та організовувати процеси, що відповідають специфіці предметної області (методології, проекту, підприємства, технології та ін.). Застосовність підходу доводиться шляхом його використання для моделювання процесу проектування систем.*

*The principles of development and application of domain-specific models of different processes of system engineering are considered in the paper. The offered approach allows to model and organise processes which correspond to specificity of a subject domain (methodology, project, enterprise, technology, etc.). Usability of the approach is proven by its application for modelling the process of systems design.*

Ключові слова: предметно-орієнтоване моделювання, онтології.

**Постановка задачі**

В області інформаційних технологій (ІТ) наразі існує множина різних методологій проектування комп'ютерних систем. Наприкінці минулого сторіччя навіть виникло поняття війни методів. Причиною є той факт, що розроблювачі нових методологій системної інженерії свідомо ігнорували загальні підходи, що завжди лежать в основі процесів розробки систем. Саме тому поява нової методології наразі зустрічається із критичними поглядами як з боку теоретичних дослідників, так і практичних розробників в області ІТ.

Останнім часом у системній інженерії виник підхід, який отримав назву комп'ютерно-орієнтованої інженерії методів (англ. Computer Aided Method Engineering, CAME) [1, 2], що можна розглянути за аналогією з комп'ютерно-орієнтованою програмною інженерією (англ. Computer Aided Software Engineering, CASE). CASE полегшує процес розробки програмних систем, CAME здійснює підтримку діяльності інженерів з розробки методів, технологічних процесів, технологій та ін. У той же час, CAME-підхід оснований на ситуативній інженерії методів (англ. Situational Method Engineering, SME) [3], головною метою якої є розробка специфічних для проекту методологій.

Але зазначимо, що визначення методології ще недостатньо для її практичного застосування у процесі розробки систем. Для досягнення мети необхідно організувати процес, що відповідає принципам даної методології. Саме тому актуальними є задачі визначення методів моделювання процесів, що відповідають методологіям предметної області (Про), а також способів організації діяльності, що відповідає структурі даних процесів.

Запропонований у статті підхід оснований на методології предметно-орієнтованого моделювання

(англ. Domain Specific Modelling, DSM) [4]. Ідея полягає у розширенні DSM шляхом надання користувачу можливості створювати не лише статичні структурні моделі ПрО, але і моделі процесів ПрО, які дозволяють організовувати відповідну діяльність.

Особливістю та перевагою DSM є розробка моделей не в технічних термінах даних й алгоритмів, а в змістовних поняттях предметної області. Ці моделі породжуються з так званої метамоделі, що, власне, слугує предметно-орієнтованою мовою (англ. *Domain Specific Language, DSL*) [5] для розробки моделей ПрО. Однак відмітимо, що методологія DSM призначена, головним чином, для розробки програмних систем, у той час як ми не обмежуємо область застосовності нашого підходу.

У статті [6] нами була розглянута концептуальна метамодель (онтологія), що дозволяє будувати статичні моделі властивостей та структури різних ПрО. Предметом розгляду в даній статті є метамоделі, що можуть бути використані для породження та застосування моделей процесів ПрО (моделювання, проектування, стандартизація та ін.). Якщо онтологічна метамодель будувалася на основі математичної теорії графів, метамодель процесів пропонується створити на базі теорії кінцевих автоматів. Іншими словами, сутність запропонованого у статті підходу полягає в побудові моделі процесу системної інженерії на основі кінцевого автомата, де умови визначають порядок переходу станів у процесі розробки системи.

Зазначимо, що використання математичних теорій як основи для побудови метамodelей дозволяє визначити наш підхід як предметно-орієнтоване математичне моделювання (англ. *Domain Specific Mathematical Modelling, DSMM*).

Запропонований підхід був практично реалізований у *web*-орієнтованому інструменті, названому нами *OpenCookbook* [6]. Застосовність DSMM у даній статті доводиться на прикладі моделювання процесу проектування систем.

### Визначення поняття методології та моделі процесу

Методологія визначається нами як упорядкований набір методів, які можуть бути застосовані до моделі предметної області з метою розв'язання виникаючих у ПрО задач. У даній статті ми будемо робити акцент на розгляді методологій проектування систем.

Помітимо, що більшість методологій проектування систем є слабко або ж взагалі не формалізованими. Наприклад, об'єктно-орієнтована методологія (ООМ), описана у відомій книзі [7], взагалі кажучи, є множиною текстових рекомендацій, що досить далекі від формальної техніки проектування систем.

У той же час відзначимо: щоб організувати відповідний методології процес, її принципи повинні бути не лише ясно зрозумілі, але і формалізовані. Саме тому є необхідним формальне визначення моделі процесу, що відповідає певній методології системної інженерії.

Грунтуючись на визначенні метамоделі моделі ПрО як графа [6], методологія може бути визначена як множина методів обробки й перетворення графів моделей ПрО.

Побудована в такий спосіб методологія використовувалася нами зокрема для розв'язання задач над моделями, що відбивали структуру програмних систем [8]. У даному підході методологія визначала структуру процесу розв'язання задачі, а саме генерації програмного коду та пошуку мінімального шляху в топології обчислювальних вузлів.

Але зазначимо, що такий підхід може застосовуватися лише у випадку достатньої формалізованості ПрО. Однак, як було зазначено нами вище, існуючі методології системної інженерії (зокрема, проектування, а також стандартизації систем) достатньо далекі від математично точного визначення.

У цьому випадку доцільність запропонованого нами підходу полягає в можливості створення на основі теорії графів візуальної метамоделі (нотації), що дозволяє користувачу задати структуру процесу предметної області у поняттях самої ПрО.

Наступний розділ статті ілюструє даний підхід.

### Породження графічних нотацій

У межах системної інженерії наразі існує кілька методологій проектування систем і відповідних їм графічних нотацій: діаграми сутність-зв'язок (англ. *Entity-Relationship Diagrams, ERD*) [9, 10], діаграми функціонального моделювання (англ. *Structured Analysis and Design Technique, SADT*) [11], діаграми потоків даних (англ. *Data-Flows Diagrams, DFD*) [12], ООМ [7] з нотацією UML (*Unified Modelling Language*) [13] та ін.

Зазначимо, що використання методології проектування систем іноді зводиться лише до застосування відповідної графічної нотації у деякому CASE інструменті. Перевагою запропонованого нами підходу є те, що користувач може визначити власну графічну нотацію (графічну метамодель), що відповідає специфіці ПрО.

Практично кажучи, всі існуючі у системній інженерії графічні нотації можуть бути визначені на основі графів. Це є результатом того факту, що граф дозволяє здійснити декомпозицію ПрО найбільш звичним для людини способом, а саме – в об'єкти (вершини графу) та їх залежності (ребра графу). Іншим аспектом є можливість атрибутизації ребер графа, виділення ієрархічних деревоподібних структур у графі та ін.

Зокрема, ERD нотація один до одного відповідає основаній на теорії графів метамоделі. В ERD сутність має множину атрибутів, що включають також її унікальне ім'я (ідентифікатор). Залежність визначається як відношення або асоціація між окремими сутностями ERD.

Для породження ERD нотації ми повинні визначити ERD-сутність як вершину графа, а ERD-відношення – як ребро графа (що власне і є метамоделлю ERD). На етапі моделювання визначаються

специфічні для ПрО атрибути сутностей. Для графічного позначення вершин і ребер *ER*-діаграм використовується бібліотека графічних символів.

Зазначимо, що *ER*-діаграми застосовуються в основному для проектування баз даних [9; 10]. *DSMM* підхід може бути використаний у контексті найбільш застосовної наразі в теорії і практиці проектування систем об'єктно-орієнтованої методології.

*ER*-діаграми також використовуються в контексті *OOM*, однак найбільш популярною є *UML* нотація [13], що має 14 типів діаграм (як для *UML v.2.3*). Всі ці типи *UML* діаграм можуть бути породжені у процесі *DSMM*. Причиною є той факт, що *OOM* розглядає ПрО (або майбутню систему) також як множину взаємозалежних об'єктів. Об'єкт визначається як частина дійсності, що має певні властивості, які можуть бути зафіксовані атрибутами.

Взагалі кажучи, відмінності *UML* діаграм полягають лише в графічних позначеннях та специфічних для ПрО атрибутах вершин і ребер графа.

Наведемо, як приклад, *UML* діаграму класів (рис. 1):

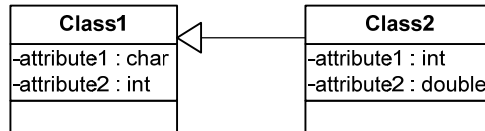


Рис. 1. Приклад *UML* діаграми класів

Але зазначмо, що для організації процесу системної інженерії недостатньо відтворити існуючу графічну нотацію (як-то *ERD*, або ж навіть *UML*), чи навіть створити нову візуальну метамодель, що відповідає специфіці ПрО. Саме тому у *DSMM* ми робимо акцент на можливості моделювання та організації відповідного специфіці ПрО процесу. Наступні розділи статті пояснюють цей підхід.

#### Моделювання процесу розробки системи

У статті [6] нами була визначена концептуальна метамодель (онтологія), що використовувалася з метою проектування систем. Базовими поняттями метамоделі є вимога, специфікація, робочий план, задача розробки, задача тестування, задача валідації, задача верифікації й ін.

Ці поняття мають атрибути й зв'язані відношеннями, тобто теорія графів може бути розглянута як мета-метамодель онтології. При цьому методи теорії графів використовувалися нами для виділення в онтологіях дерев, що відбивають різні аспекти концептуальної моделі майбутньої системи (наприклад, визначення послідовності понять, що слідує від вимоги до опису архітектурного елементу, що повинен втілити цю вимогу в життя).

Але зазначимо, що методи теорії графів є застосовними до вже існуючої онтології – концептуальної моделі майбутньої системи. При цьому відкритим залишається питання організації процесу проектування системи.

Важливим методологічним принципом нашого підходу до проектування систем є вислів: щоб спроекувати правильну систему, система повинна бути спроекувана правильно. Для «правильного проектування» необхідно визначити способи моделювання та управління процесом розробки системи.

Якщо для побудови онтологічної метамоделі майбутньої системи ми використовували теорію графів, то для моделювання процесу розробки ця метамодель повинна бути розширена положеннями теорії кінцевих автоматів. У даному контексті модель процесу розробки системи визначається як кінцевий автомат, який будується шляхом накладення обмежень на переходи між різними етапами проектування системи. Наприклад, етап архітектурного моделювання може мати місце тільки після узгодження всіх специфікацій архітектури системи.

Порядок переходів кінцевого автомату (процесу розробки системи) визначається логічними умовами, покладеними на величини атрибутів екземплярів понять онтології системної інженерії. Наприклад, всі поняття запропонованої онтології мають атрибут статус (*status*) із набором можливих значень «в роботі» («*in work*»), «схвалене» («*approved*»), «не застосовно» («*not applicable*»). Зв'язком між поняттями «вимоги» (*requirement*) і «специфікації» (*specification*) є логічна імплікація:

$$\forall((\text{Requirement.Status} = \text{Approved}) \vee (\text{Requirement.Status} = \text{Not applicable})) \rightarrow \text{Specification.Status} = \text{Approved} \quad (1)$$

Тобто специфікація може бути схвалена, якщо відповідні вимоги були схвалені (або ж незастосовні). Таким чином, перехід до наступного рівня визначення системи (рівня специфікацій) можливий лише після схвалення попереднього (рівня вимог).

Таким чином, логічні умови, що покладені на відношення понять онтології проектування систем, визначають порядок зміни станів системи між різними кроками в її розробці (наприклад, рис. 2). Ці зміни станів і є, власне, процесом системної інженерії.

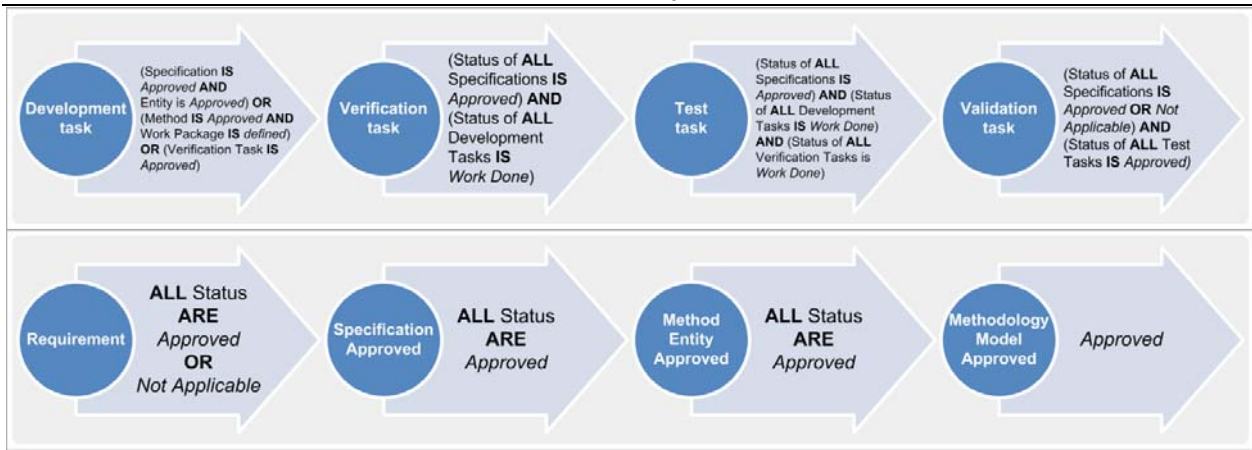


Рис. 2. Переходи станів моделі процесу розробки системи

Інша особливість *DSMM* – зв’язування декларативних умов з певними імперативними діями. Якщо логічна формула є істиною, викликається відповідній умові метод (який, у простішому випадку, встановлює значення атрибута *status* в *approved*). Як відзначалося у статті [6], процес побудови онтологічної моделі майбутньої системи є, власне, створенням екземплярів (інстаціацією) абстрактних понять метамоделі (тобто вимог, специфікацій, архітектурних елементів та ін.). Тому в загальному випадку умови визначають не лише можливі зміни атрибутів понять онтології, але і можливість створення (знищення) екземплярів понять, тобто структуру концептуальної моделі системи.

У загальному випадку модель процесу системної інженерії визначає, як розробити систему правильним способом. На рис. 3 зображені можливі переходи між різними етапами у визначенні системи. Зазначимо, що ці етапи є також поняттями онтології проектування систем (вимога, специфікація та ін.).

**Модельовання процесів структурного проектування**

Розглянемо застосовність *DSMM* підходу на прикладі модельовання процесів, що відповідають методології структурного проектування.

Діаграми структурного проектування були створені в середині 1960-х, коли Дуглас Т. Росс запропонував спеціальну технологію *SADT* (*Structured Analysis and Design Technique*). Пізніше *SADT* стала основою для методології *IDEF* [14]. Далі в 1981 був створений стандарт *IDEF0* (*Icam DEFinition*) у рамках програми інтеграції комп’ютерних та індустріальних технологій *ICAM* (*Integrated Computer Aided Manufacturing*) [15].

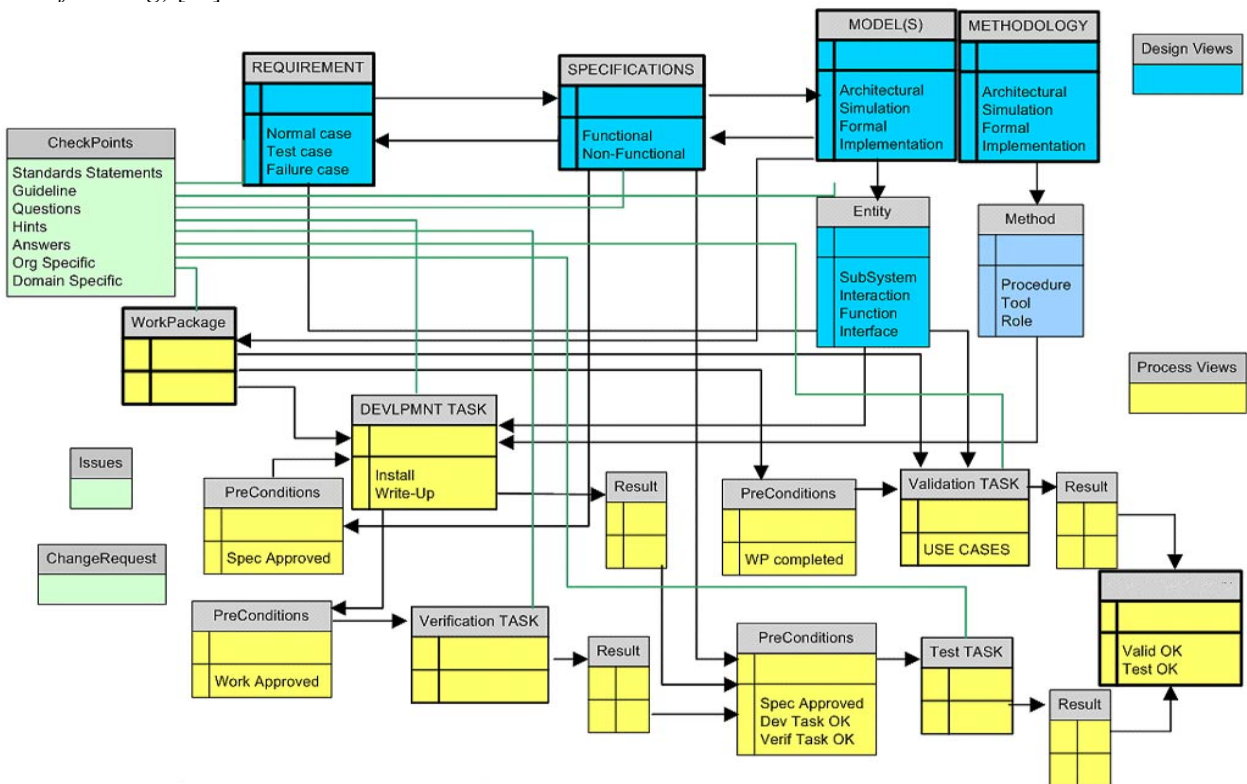


Рис. 3. Переходи між різними станами у визначенні системи

Методологія *IDEF-SADT* включає набір методів, принципів і процедур для розробки функціональних моделей систем у деякій ПрО. Функціональна модель *SADT* відображає структуру процесів

як системи у цілому, так й її окремих підсистем. *IDEF* розглядає систему як множину функцій, які перетворюють вхідні дані у вихідні.

Для моделювання процесів, що відповідні методології *IDEF-SADT*, кожен елемент структурованої у виді графа онтологічної моделі розглядається нами як процес. Процес бере участь (або використовується) у виконанні деякої функції або операції (перетворення, обробки, формування й т.ін.). Функція є логічною композицією операцій (дій), які перетворюють вхід у вихід. Послідовність взаємозалежних входів і виходів функцій становить процес. Функції можуть також генерувати об'єкти будь-якої природи. Взагалі кажучи, функції можуть виконуватися в будь-якій послідовності (наприклад, циклічній).

Порядок виконання функцій залежить від різних умов, як, наприклад, зовнішніх та внутрішніх подій. Саме події викликають виконання функцій, які, у свою чергу, змінюють умови й формують нові події, поки процес не буде завершений. Таким чином, фіксація послідовності подій є одним із способів моделювання та виконання процесу.

Структурний системний аналіз (*SSA*) [16] є методом моделювання систем, що починається з найбільш загального опису з наступною деталізацією окремих аспектів властивостей і поведінки системи.

Обмеження *SSA* стають видимими, коли необхідно деталізувати подання моделі системи, наприклад, крім статичних відношень відтворити поведінку або ж функціонування окремих вузлів системи. З цією метою в межах *SSA* використовуються діаграми потоків даних *DFD*.

Для моделювання майбутньої системи відповідно до структурної методології спочатку треба представити у формі графів найбільш загальні й абстрактні функції моделі системи. Ці графи повинні мати обмежене число вузлів на кожному рівні ієрархії, що відбивають лише найбільш істотні елементи системи.

Специфічною особливістю структурного підходу є декомпозиція системи у функціональні підсистеми. Така декомпозиція ітеративно застосовується до підсистем, поки вони не стають досить простими, щоб бути змодельованими. У цьому випадку модель зберігає цілісність, в якій всі підсистеми (вершини) зв'язані ребрами, що відбивають відношення ПрО. Відмітимо, що при розробці системи знизу нагору, тобто від окремих елементів до цілої системи, її цілісність може бути загублена.

Разом з декомпозицією системи у вершини графа (підсистеми) і ребра графа (функціональні залежності) треба використовувати метод ієрархічного упорядкування. Це означає упорядкування елементів графу моделі системи в деревоподібні структури за певним критерієм, з послідовною деталізацією на кожному рівні ієрархії.

Взагалі система декомпозується у множину підсистем, представлених у формі "чорних ящиків" із множиною входів і виходів. Така декомпозиція з наступною організацією цих підсистем в ієрархічні деревоподібні структури є основним методом структурного моделювання.

Моделювання систем відповідно *SSA* потребує розробки наступних метамodelей з відповідними графічними нотаціями:

- об'єктної структури, що відбиває архітектуру системи;
- функціональної структури, що відбиває дії й взаємодію об'єктів;
- управляючої структури, що відбиває поведінку системи.

Звернемо також увагу на основне розходження між структурною й об'єктною методологією, що полягає в способі поєднання функцій і даних. В *OOM* дані й методи репрезентують єдність (т.зв. принцип інкапсуляції). Однак, структурна й об'єктна методологія не є семантично протилежними й можуть бути об'єднані в єдину методологію за допомогою *DSMM*. Цю задачу ми залишаємо для наших майбутніх досліджень.

### Висновки

У статті розглянуті принципи моделювання та організації процесів системної інженерії на основі інформаційної технології предметно-орієнтованого математичного моделювання. Доцільність застосування *DSMM* полягає в теоретичній і практичній можливості створення modelей, що найбільш повно відповідають запитам ПрО (підприємства, проекту, технології та ін.).

*DSMM* підхід оснований на методології предметно-орієнтованого моделювання (*DSM*). Відмінність полягає у розширенні метамodelей, що дозволяє користувачеві створювати не лише *modelі* ПрО, але і організовувати специфічні для ПрО процеси.

Іншим аспектом *DSMM* є побудова метамodelей на основі математичних теорій та використання їх методів для розв'язання задач над modelями ПрО. Так, теорія графів використовувалася нами для побудови онтологічних modelей ПрО, а різні способи обходу графу визначали структуру розв'язку специфічної для ПрО задачі.

Для побудови modelей процесів розробки систем запропоновано метамodelь, що основана на теорії кінцевих автоматів. У цьому контексті модель процесу проектування майбутньої системи розглядається нами як кінцевий автомат, що визначає порядок переходу станів у процесі розробки системи.

Наші майбутні дослідження будуть присвячені подальшій розробці й формалізації modelей процесів ПрО. Це дозволить розширити класи задач, що можна розв'язати у межах *DSMM*.

### Література

1. Ajantha Dahanayake. Computer-Aided Method Engineering: Designing CASE Repositories for the 21st

Century. – IGI Global. – 2001. – 252 p.

2. Ali Niknafs and Raman Ramsin. Computer-Aided Method Engineering: An Analysis of Existing Environments // LNCS. – Springer. – Volume 5074. – 2008. – 525-540 pp.

3. Jolita Ralyte and Sjaak Brinkkemper. Situational Method Engineering: Fundamentals and Experiences: Proceedings of the IFIP WG 8.1 Working Conference, 12-14 September 2007, Geneva, Switzerland. – Springer US. – 380 p.

4. Steven Kelly and Juha-Pekka Tolvanen. Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Pr. – 2008. – 427 p.

5. Richard C. Gronback. Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit. Addison-Wesley Professional. – 2009. – 736 p.

6. Vitaliy Mezhujev, Eric Verhulst, and Bernhard H.C. Sputh. Interacting Entities Modelling Methodology for Robust Systems Design // Papers of the Second International Conference on Advances in System Testing and Validation Lifecycle. – CPS publishing. – 2010. – 75– 80 pp.

7. Grady Booch, Robert A. Maksimchuk, Michael W. Engel, and Bobbi J. Young. Object-Oriented Analysis and Design with Applications (3rd Edition). – Addison-Wesley Professional. – 2007. – 720 p.

8. Межуев В. И. Предметно-ориентированное моделирование распределенных параллельных приложений реального времени / В. И. Межуев // Системи обробки інформації. – 2010 – Вип. 5 (86). – С. 98 – 103.

9. Sikha Bagui and Richard Earp. Database Design Using Entity-Relationship Diagrams (Foundations of Database Design). Auerbach Publications. – 2003. – 264 p.

10. B. Thalheim. Entity-Relationship Modeling: Foundations of Database Technology. – Springer Berlin Heidelberg. – 2010. – 640 p.

11. David A. Marca and Clement L. McGowan. IDEF0 and SADT: A Modeler's Guide. – OpenProcess, Inc. – 2005. – 392 p.

12. DiCerto J. J. Planning and preparing data-flow diagrams. Hayden Book Co. – 1964. – 90 p.

13. Russ Miles and Kim Hamilton. Learning UML 2.0. – O'Reilly Media. – 2006. – 288 p.

14. <http://www.idef.com/>

15. <http://www.icam.com>

16. Introduction to Methodologies and SSADM [Електронний ресурс]. – Режим доступу: <http://www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4.html>

Надійшла 18.9.2010 р.

УДК 389.14: 006.15.7

В.Т. КОНДРАТОВ

Институт кибернетики им. В.М. Глушкова НАН Украины

## ТЕОРИЯ МЕТРОЛОГИЧЕСКОЙ НАДЕЖНОСТИ: ПРИМЕНЕНИЕ НЕЛИНЕЙНОГО РЕГРЕССИОННОГО АНАЛИЗА ДЛЯ ПРОГНОЗИРОВАНИЯ И ОПРЕДЕЛЕНИЯ ВРЕМЕНИ НАРАБОТКИ НА МЕТРОЛОГИЧЕСКИЙ ОТКАЗ

*В статье впервые показана возможность использования нелинейного регрессионного анализа в теории метрологической надежности. Приведены основные понятия и определения, связанные с нелинейным регрессионным анализом.*

*In the article possibility of the use of nonlinear regressive analysis is first rotined in the theory of metrology reliability. Basic concepts and determinations, related to the nonlinear regressive analysis, are resulted.*

Ключевые слова: надежность, метрология, регрессионный анализ.

### Введение

Задача определения значения прогнозного времени наработки на метрологический отказ (МО) была, есть и остается актуальной до тех пор, пока не будут синтезированы вероятностно-физические модели (ВФ-модели) МО для всех типов средств измерений (СИ).

Решение указанной задачи стало возможным благодаря разработке правил синтеза гибких многопараметровых функций распределения (ФР) нормируемых погрешностей в течение времени наработки на МО [1]. Они обуславливают синтез ФР с независимым параметром масштаба и связью его со значениями погрешностей, полученных при аттестации СИ или ввода его в эксплуатацию. Свойства и особенности некоторых ФР описаны в работах [1– 10].

За 2007– 2009 гг. нами синтезировано три вида ФР, характеризующих взаимосвязь метрологических характеристик (МХ), в частности, нормируемых погрешностей, с таким параметром метрологической надежности (МН), как время наработки СИ на МО. Однако эти ФР пока не привязаны к СИ соответствующего типа.

В 2009– 2010 гг. удалось решить метрологическую задачу применения регрессионного анализа для динамического прогнозирования и определения времени наработки на МО и разобраться в тонкостях