УДК 004.942:519.876.5

О. П. ДОРЕНСЬКИЙ, О. А. СМІРНОВ, д-р техн. наук, професор,
Кіровоградський національний технічний університет

ФОРМАЛІЗАЦІЯ ПРОЦЕСУ ЗМІНИ СТАНІВ ПРОГРАМНИХ ОБ'ЄКТІВ СКЛАДНИХ СИСТЕМ НА ОСНОВІ СКІНЧЕННИХ АВТОМАТІВ МУРА

У пропонованій статті модель станів представників класу об'єктів програмного забезпечення складних систем, розроблюваного на основі об'єктно-орієнтованого підходу, розглянуто з погляду абстрактного синтезу скінченного автомата. Обґрунтовано й використано ініціальний скінченний автомат Мура. Здійснено перетворення моделі станів у скінченний автомат. Розкрито сутність формалізації послідовності подій здобутої скінченно-автоматної моделі поведінки представників класу об'єктно-орієнтованого програмного забезпечення, а також особливості формалізації результатів її функціонування на стадії проектування програмних систем.

Ключові слова: програмне забезпечення; об'єктно-орієнтований підхід; проектування; програмний об'єкт; стан; поведінка; модель; скінченний автомат Мура.

Вступ

Програмне забезпечення (ПЗ) сучасних інфокомунікаційних систем, автоматизованих систем керування, прикладних додатків тощо характеризується складністю [1], інформаційною, структурною та алгоритмічною надмірністю [2]. Водночас до такого ПЗ висувається низка жорстких вимог щодо надійності, високопродуктивності, гнучкості, функціональності, супроводжуваності, зручності, переносимості [3], а також уможливлення заходів із його вдосконалення, масштабування, нарощування функціональності, швидкого розроблення.

Виконання зазначених вимог і характеристик програмних систем досягають завдяки застосуванню об'єктно-орієнтованого підходу (ООП) [1; 4–6]. На відміну від традиційного підходу, ООП передбачає, що увагу приділяють як інформації, так і поведінці системи, а предметну область розбивають на деяку множину незалежних сутностей — об'єктів [4]. Проте навіть застосування ООП не гарантує уникнення, виявлення та усунення алгоритмічних і системних помилок у ПЗ, абсолютна більшість яких виникає на ранніх стадіях життєвого циклу ПЗ, передусім на етапі проектування [4–7]. Загалом розроблення складних програмних систем неможливе без використання моделей — комплексної моделі об'єктно-орієнтованого ПЗ, її статичного і динамічного компонентів [5], тобто без моделювання й тестування кожної системи під час її проектування.

Аналіз останніх досліджень і публікацій

Процес проектування об'єктно-орієнтованого ПЗ (ООПЗ) полягає в побудові, послідовному дослідженні й контролі сукупності узгоджених моделей, які описують функції програмної системи мовою об'єктно-орієнтованого моделювання, а також візуалізуються у вигляді діаграм [1; 4; 6]. Комплекс узгоджених об'єктно-орієнтованих моделей у сукупності являє собою комплексну (інтегровану) модель ООПЗ, яка складається зі статичного і динамічного компонентів [5].

У [5] задля формалізації основних понять мови об'єктно-орієнтованого моделювання на рівні її метамодельного подання введено новий вид опису метамоделі в термінах теорії множин, проаналізовано комплексну модель ООПЗ, її рівні та компоненти. При цьому визначено основні концептуальні одиниці, їхні властивості та функції. Усе це становить підґрунтя для математичної формалізації об'єктно-орієнтованих моделей ПЗ. Так, на базі теорії множин здійснено формалізацію в термінах об'єктно-орієнтованого підходу основних концептуальних одиниць статичного компонента (об'єкт, клас, атрибут, зв'язок, асоціація) і концептуальних одиниць динамічного компонента (прецедент, сценарій, стан, подія, діяльність, перехід) комплексної моделі об'єктно-орієнтованого ПЗ, а також процесів абстрагування зазначених одиниць.

У [7] на основі результатів дослідження [5] застосовано стратегію синтезу об'єктно-орієнтованих моделей у термінах об'єктно-орієнтованого підходу й мови об'єктно-орієнтованого моделювання. Зрештою здійснено синтез структури комплексної моделі ООПЗ, її статичного компонента (модель класів), динамічного компонента в межах колективної та індивідуальної поведінки основних концептуальних одиниць (модель прецедентів, моделі станів, моделі діяльностей, моделі послідовностей, моделі кооперацій), обґрунтовано доцільність опису моделей у вигляді відповідних семантичних мереж або скінченних автоматів.

Постановка задачі

Істотна особливість об'єктно-орієнтованого ПЗ складних систем під час функціонування полягає в обміні великою кількістю інформації між програмними об'єктами, зокрема повідомленнями між представниками класу програмних об'єктів

та класами. На стадії проектування цей процес описується моделями поведінки програмних об'єктів ООПЗ [8], складової динамічного компонента комплексної моделі ООПЗ [5; 7], зміни їхніх станів. Отже, постає актуальна науково-прикладна задача математичної формалізації процесу зміни станів представників класу програмних об'єктів об'єктно-орієнтованого ПЗ. Її розв'язок забезпечить можливість застосування математичних методів задля подальшого синтезу логічних і тестових моделей поведінки програмних об'єктів ООПЗ, що має на меті їх застосування на стадії проектування програмних систем.

Задля автоматизації побудови моделей під час синтезу структури динамічного компонента комплексної моделі ОППЗ складних систем на етапі проектування [7; 14] процес побудови моделі стану представників класу об'єктів ООПЗ розглядається з погляду абстрактного синтезу скінченного автомата. Такий підхід цілком обґрунтований і доцільний, оскільки модель стану представників класу описується за допомогою скінченного автомата [9]; методи абстрактного синтезу скінченних автоматів забезпечують виявлення закону функціонування автомата, установлення обсягу його пам'яті та формальний опис такого автомата одним зі стандартних способів (таблиці переходів і графи станів).

Модель станів у нотації Харела поєднує в собі деякі аспекти автоматів Мура та автоматів Мілі, а також концепції, які стосуються вкладених і паралельних станів. Тому задля використання методів синтезу скінченних автоматів на початковому етапі побудови моделі станів представників класу об'єктів ООПЗ слід звести модель зміни станів до одного з видів скінченних автоматів: автомата Мілі або атомата Мура. При цьому після її побудови вона може бути доопрацьована вручну, що забезпечить використання всіх можливостей нотації Харела.

В ініціальному скінченному автоматі Мура вихідні сигнали пов'язані зі станами, що відповідає поданню моделі станів представників класу програмних об'єктів комплексної моделі ООПЗ. Отже, для її побудови є сенс і всі підстави скористатись зазначеним автоматом. При цьому скінченні автомати надалі слід розглядати з погляду вивчення їх можливостей у термінах множини слів, з якими вони працюють.

Мета статті — ***математична формалізація процесу зміни станів об'єктів ООПЗ складних систем за допомогою формального апарату скінченних автоматів Мура***. Розв'язати цю науково-прикладну задачу можна перетворенням моделі станів об'єктів ООПЗ до вигляду скінченного автомата другого роду (автомата Мура), визначенням закону функціонування автомата та формалізацією послідовності подій здобутої скінченно-автоматної моделі й результатів її функціонування.

1. Перетворення моделі станів представників класу об'єктів ООПЗ до скінченного автомата Мура

Скінченний автомат Мура

$$A=(X,Y,V,F_{XV},F_{VY},\upsilon_0,V')$$

задається як сукупність понять, що включає в себе скінченний вхідний алфавіт (множину вхідних сигналів) $X=\{\chi_0,\chi_1,...,\chi_n\}$, скінченний вихідний алфавіт (множину вихідних сигналів) $Y=\{\upsilon_0,\upsilon_1,...,\upsilon_m\}$, скінченну множину станів $V=\{\upsilon_0,\upsilon_1,...,\upsilon_{sc}\}$, характеристичну функцію переходів $F_{XV}: X\times V\to V$, характеристичну функцію виходів (зсунуту функцію виходів) $F_{VY}: V\to Y$, початковий стан скінченного автомата $\upsilon_0\in V$, а також множину заключних станів $V'\subseteq V$. При цьому закон функціонування зазначеного автомата набирає такого вигляду:

$$\upsilon_j=F_{XV}(\chi,\upsilon_i),\ \ \upsilon_i=F_{VY}(\upsilon_i).$$

Існує кілька способів задання скінченних автоматів [9–12]. Для подання автомата A використовуватимемо відмічену таблицю переходів і відповідний їй граф переходів. Відмічена таблиця переходів відповідає характеристичій функції переходів $F_{XV}(\chi,\upsilon_i)$, її клітинки заповнюються номерами станів υ_j, в які переходить автомат під час впливу χ у стані υ_i. При цьому над кожним станом автомата, який позначає певний стовпець таблиці, міститься відповідний цьому стану вихідний сигнал $\upsilon_i=F_{VY}(\upsilon_i)$. Отже, в автоматі A кожний стан υ відзначено відповідним йому вихідним сигналом υ.

Граф переходів автомата Мура є орієнтованим мультиграфом, який задається двома неперетинними множинами: множиною вершин — стани автомата A; множиною дуг — вхідні сигнали автомата A. Цей граф задається також функціями інцидентності, кожна з яких визначає для кожної дуги її початкову та кінцеву вершину.

Життєвий цикл представника деякого класу ООПЗ $sc\in SC$ [5], який описується моделлю станів [4; 7], перетворюється у скінченний автомат Мура A згідно з таким алгоритмом [13].

1. Кожному елементу множини станів SS_{sc} моделі станів MS_{sc} класу програмних об'єктів $sc\in SC$ відповідає **єдиний елемент множини станів** V скінченного автомата Мура A, тобто визначено ***бієктивне відображення***:

$$F_{VSs}: V\to SS_{sc}. \qquad (1)$$

2. Кожному елементу множини подій SE_{sc} моделі станів MS_{sc} класу програмних об'єктів $sc\in SC$ відповідає **єдиний елемент множини вхідних сигналів** X скінченного автомата Мура A, тобто визначено ***бієктивне відображення***

$$F_{XSe}: X\to SE_{sc}. \qquad (2)$$

3. Кожному елементу множини діяльностей SH_{sc} моделі станів MS_{sc} класу програмних об'єктів $sc \in SC$ відповідає **єдиний елемент множини вихідних сигналів** Y автомата A, тобто визначено ***бієктивне відображення***

$$F_{YSh}: Y \to SH_{sc}. \quad (3)$$

4. Кожній впорядкованій парі

$$((se, ss_i), ss_j) \in F_{SesSs}: SE_{sc} \times\times SS_{sc} \to SS_{sc}$$

моделі станів MS_{sc} відповідає **єдина впорядкована пара** $((\chi, \upsilon_i), \upsilon_j) \in F_{XV}: X \times V \to V$ скінченного автомата Мура A, тобто для будь-яких $sc \in SC$ та $ss \in SS_{sc}$ виконується умова

$$F_{SesSs}(F_{XSe}(\chi), F_{VSs}(\upsilon)) = F_{VSs}(F_{XV}(\chi, \upsilon)). \quad (4)$$

5. Кожній впорядкованій парі

$$(ss, sh) \in F_{SsSh}: SS_{sc} \to SH_{sc}$$

моделі станів MS_{sc} відповідає **єдина впорядкована пара** $(\upsilon, \upsilon) \in F_{VY}: V \to Y$ автомата Мура A, тобто для будь-яких елементів $sh \in SH_{sc}$ та $ss \in SS_{sc}$ виконується умова

$$F_{SsSh}(F_{VSs}(\upsilon), F_{YSh}(\upsilon)) = F_{YSh}(F_{VSs}(\upsilon)). \quad (5)$$

6. Початковому стану створення ss_0 моделі станів MS_{sc}, $ss_0 \in SS_{sc}$, відповідає стан $\upsilon_1 \in V$ скінченного автомата A, тобто згідно з виразом (1)

$$F_{VSs}(\upsilon_0) = ss_0. \quad (6)$$

7. Кожному елементу множини станів $SS_{sc}^{sac} \subset SS_{sc}$ моделі станів MS_{sc} класу об'єктів $Sc \in SC$ відповідає **єдиний елемент множини станів** $V' \subseteq V$ автомата A, тобто визначено ***бієктивне відображення***

$$F_{V'Ss}: V' \to SS_{sc}^{sac}. \quad (7)$$

Отже, у результаті перетворення моделі станів MS_{sc} представників класу об'єктів ООПЗ доходимо такого висновку: **трійка взаємно однозначних відображень згідно з виразами (1), (2), (3) у разі виконання умови (4) визначає *ізоморфізм* MS_{sc} *у автомат* A.** Тобто модель станів MS_{sc} і автомат A відрізняються один від одного лише позначеннями вхідних та вихідних сигналів і станів. Звідси випливає, що ***операція перетворення моделі станів у ізоморфний їй скінченний автомат зводиться до перепозначення елементів вхідного і вихідного алфавітів та множини станів скінченного автомата в термінах моделі станів класу об'єктів ООПЗ.***

Отже, автомат Мура, який використано для початкового формалізованого подання моделі станів MS_{sc}, тобто скінченно-автоматну модель поведінки деякого класу об'єктів ООПЗ, можна задати сукупністю таких понять:

$$A_{sc} = (SS_{sc}, SE_{sc}, SH_{sc}, F_{SeSs}, F_{SsSh}, ss_0, SS_{sc}^{s}) \quad (8)$$

Автомат A_{sc} являє собою частковий або не повністю визначений автомат, оскільки функція переходів F_{SeSs} є не повністю визначеною для всіх наборів із множини $(SE_{sc} \times SS_{sc})$.

2. Формалізація послідовності подій скінченно-автоматної моделі поведінки програмного об'єкта

Для автомата A_{sc} характеристична функція переходів F_{SeSs} може визначатись не тільки на скінченному вхідному алфавіті SE_{sc}, який складається з усіх подій класу $sc \in SC$, а й на множині всіх можливих трасувань послідовностей подій класу об'єктів ООПЗ у алфавіті SE_{sc}. Для будь-якого трасування $ste \in STE_{sc}$, що являє собою вхідне слово ініціального часткового скінченного автомата Мура A_{sc} у алфавіті SE_{sc}, де $ste = se_1 se_2 \ldots se_n$, справджується таке твердження:

якщо $F_{SeSs}(ste, ss)$ визначена, то

$$F_{SeSs}(ste\, se_i, ss) \cong \cong F_{SeSs}(se_i, F_{SeSs}(ste, ss)),$$

де « ≅ » означає, що обидві функції є або одночасно невизначеними, або визначеними й рівними між собою;

якщо $F_{SeSs}(ste, ss)$ невизначена, то $F_{SeSs}(ste\, se_i, ss)$ є невизначеною для всіх $ss \in SS_{sc}$.

Довільну множину слів у будь-якому алфавіті називають ***подією*** у відповідному алфавіті, сам же алфавіт — ***вхідним алфавітом відповідної події.***

Зазначений термін є традиційним і в теорії автоматів [11; 12]. Множину трасувань послідовностей подій класу об'єктів ООПЗ STE_{sc} також можна розглядати як деяку подію у вхідному алфавіті SE_{sc} (множину подій деякого класу об'єктів ООПЗ) скінченного автомата A_{sc}, де кожне трасування послідовності $ste \in STE_{sc}$ є вхідним словом у алфавіті SE_{sc}.

Задля уникнення плутанини щодо події як ***множини слів у вхідному алфавіті*** та як ***абстракції її представників у комплексній моделі*** ООПЗ, надалі використовуватимемо таку термінологію:

- подію як множину слів у вхідному алфавіті називатимемо ***подією в довільному вхідному алфавіті***;
- подію як абстракцію її представників у комплексній моделі ООПЗ називатимемо ***подією класу програмних об'єктів***.

Розглянемо деякі особливості подання подій у SE_{sc}.

- ***Елементарною подією*** в алфавіті $SE_{sc} = \{se_1, se_2, \ldots, se_n\}$ потужністю n називають $n + 1$ одноелементну подію в цій множині, яка відповідає подіям класу об'єктів $cs \in CS$ ООПЗ $se_1, \ldots, se_n$ та порожньому слову se^0;
- ***одноелементною подією*** в алфавіті SE_{sc} називають подію в цій множині, яка складається з одного слова, тобто будь-яке трасування послідовностей подій класу об'єктів ООПЗ $ste \in STE_{sc}$ є одноелементною неелементарною подією (наприклад, $ste = se_1 se_2 se_3 \ldots se_m$);
- ***скінченною подією*** називають будь-яку подію в довільному скінченному вхідному алфавіті, яка складається зі скінченної множини слів.

Наприклад, множина всіх трасувань послідовностей подій STE_{sc} деякого класу об'єктів $cs \in CS$ ООПЗ являє собою скінченну подію.

Довільна скінченна подія $SFE_{sh} \in \sigma(STE_{sc})$ у алфавіті SE_{sc}, де $\sigma(STE_{sc})$ — множина всіх підмножин множини STE_{sc}, подається у скінченно-автоматній моделі поведінки A_{sc} класу $cs \in CS$ ООПЗ деякою діяльністю $sh \in SH_{sc}$, якщо $\forall ste \in SFE_{sh} F_{SsSh}(ste, ss_0) = sh$, тобто якщо SFE_{sh} складається з усіх тих і тільки тих послідовностей подій класу (трасування послідовностей подій класу програмних об'єктів), які викликають появу послідовностей діяльностей класу об'єктів ООПЗ, котрі закінчуються однією і тією самою діяльністю *sh*.

Множина, яка складається з подій у алфавіті SE_{sc} для всіх $sh \in SH_{sc}$ автомата A_{sc}, — канонічна множина подій у алфавіті SE_{sc} цієї скінченно-автоматної моделі.

Для будь-якої скінченної множини подій у будь-якому скінченному вхідному алфавіті тоді і тільки тоді існує скінченний автомат, що подає кожну з подій деякою множиною своїх вихідних сигналів, коли всі події цієї множини є регулярними [5].

Отже, у вхідному алфавіті SE_{sc} подій класу $cs \in CS$ ООПЗ визначимо таке:

♦ усі події $se_1,\ldots,se_n$ класу $cs \in CS$ та порожнє слово se^0 є ***регулярними виразами***;

♦ будь-яке трасування послідовностей подій класу об'єктів ООПЗ $ste \in STE_{sc}$ є ***регулярним виразом***, що подається у вигляді добутку елементарних подій у вхідному алфавіті SE_{sc} подій класу $cs \in CS$:

$$rv_{ste} = se_{i_1} se_{i_2} \ldots se_{i_m};$$

♦ множина всіх трасувань послідовностей подій класу програмних об'єктів STE_{sc} для будь-якого класу $cs \in CS$ ООПЗ є ***регулярним виразом***, що подається у вигляді диз'юнкції відповідних трасувань послідовностей:

$$rv_{ste_1} \vee rv_{ste_2} \vee \ldots \vee rv_{ste_n}.$$

3. Особливості формалізації результатів функціонування скінченно-автоматної моделі

Функціонування довільного скінченного автомата полягає в реалізації деякого відображення множини слів вхідного алфавіту (події в даному алфавіті) у множину слів вихідного алфавіту. Під час формалізації моделі переходів станів класу скінченним автоматом Мура A_{sc} зазначене відображення реалізується в такий спосіб.

Під час трасування послідовностей подій класу об'єктів ООПЗ класу *ste* в алфавіті подій класу SE_{sc} події одна за одною надходять на вхід скінченного автомата A_{sc}, який заздалегідь установлено в початковий стан. Скінченна послідовність подій класу, яка виникла в результаті цього, на основі закону функціонування автомата однозначно зумовлює появу певної скінченної послідовності діяльностей класу *sch*. Визначаючи таким чином кожному трасуванню подій *ste* відповідну йому послідовність діяльностей *sch*, дістаємо відображення множини всіх трасувань у скінченному алфавіті подій класу SE_{sc} в множину послідовностей діяльностей класу в скінченному алфавіті SH_{sc}.

Побудоване так відображення позначимо F^A_{SteSch}. Воно є ***автоматним відображенням***, індуційованим A_{sc}, і має задовольняти сформульовані далі умови автоматності [11].

• Відображення F^A_{SteSch} має здійснювати однозначну відповідність (у загальному випадку часткову) заданої множини трасувань послідовностей подій класу об'єктів ООПЗ у скінченному алфавіті подій класу SE_{sc} у певну множину послідовностей діяльностей класу в скінченному алфавіті SH_{sc} цього класу.

• Область визначення відображення F^A_{SteSch} має задовольняти умову повноти, тобто разом із будь-яким трасуванням послідовностей подій класу об'єктів ООПЗ *ste*, яке в ній міститься, вона має також включати в себе початкові відрізки цього трасування.

• Відображення F^A_{SteSch} ***має зберігати довжину слова***. Це означає, що довжина будь-якого трасування послідовностей подій класу об'єктів ООПЗ *ste* вхідного алфавіту подій класу SE_{sc}, на якому це відображення визначено, має бути така сама, як і довжина його образу.

• Відображення F^A_{SteSch} має переводити будь-який початковий відрізок трасування послідовностей подій класу об'єктів ООПЗ *ste*, на якому його визначено, у відповідний початковий відрізок образу.

Автоматне відображення F^A_{SteSch} можна також задати за допомогою подій, скориставшись аналогією до подання подій у скінченному автоматі. При цьому згідно з [11] канонічна множина подій автомата A_{sc} збігатиметься з канонічною множиною подій для індуційованого цим автоматом відображення F^A_{SteSch} за припущення, що події цієї множини позначено відповідними літерами вихідного алфавіту заданого відображення. Множина подій, яка задає автоматне відображення F^A_{SteSch} і задовольняє умови автоматності [11; 12], є ***автоматною множиною подій***.

Висновки

Згідно з результатами формалізації основних концептуальних одиниць статичного компонента (об'єкт, клас, атрибут, зв'язок, асоціація) та концептуальних одиниць динамічного компонента (прецедент, сценарій, стан, подія, діяльність, перехід) комплексної моделі об'єктно-орієнтованого програмного забезпечення, процесів їх абстра-

гування, а також синтезу структури зазначеної моделі, ***обґрунтовано скінченно-автоматне подання моделі станів програмних об'єктів складних систем і в термінах формального апарату скінченних автоматів Мура запропоновано математичну формалізацію процесу зміни станів представників класів об'єктів (моделі станів) програмного забезпечення складних систем, що розробляється відповідно до об'єктно-орієнтованого підходу.***

Модель станів програмних об'єктів розглянуто з погляду абстрактного синтезу скінченного автомата. Задля цього обґрунтовано й використано ініціальний скінченний автомат Мура, в якому вихідні сигнали пов'язані зі станами, що повною мірою відповідає побудові зазначеної моделі.

Перспектива подальших розвідок вбачається в застосуванні здобутих результатів дослідження задля синтезу моделей поведінки програмних об'єктів ООПЗ на етапі проектування складних програмних систем.

Використання розглядуваних моделей на ранніх стадіях життєвого циклу програмних систем забезпечить виявлення і, відповідно, усунення системних та алгоритмічних помилок при їх побудові, що зрештою істотно підвищить якість і надійність таких систем.

Література

1. ***Інженерія*** *програмного забезпечення: навч. посібник / [О. А. Смірнов, О. В. Коваленко, Є. В. Мелешко та ін.] — К.: РВЛ КНТУ, 2013.— 409 с.*

2. ***Різник, В.*** *Проблема подолання надмірності в інформаційних технологіях / В. Різник // Комп'ютерні науки та інформаційні технології.— 2010. — № 672.— С. 341–346.*

3. ***Поморова, О. В.*** *Сучасні проблеми оцінювання якості програмного забезпечення / О. В. Поморова, Т. О. Говорущенко// Радіоелектронні і комп'ютерні системи: якість, надійність та ресурсозбереження для апаратних і програмних засобів.— 2013.— № 5 (64).— С. 319–327.*

4. ***Дудзяний, І. М.*** *Об'єктно-орієнтоване моделювання програмних систем: навч. посібник / І. М. Дудзяний.— Львів: Видавничий центр ЛНУ ім. І. Франка, 2007.—108 с.*

5. ***Смирнов, А. А.*** *Математическая формализация процесса проектирования объектно-ориентированного программного обеспечения информационных систем / А. А. Смирнов, А. П. Доренский // Информационные технологии и системы в управлении, образовании, науке: монография под ред. проф. В. С. Пономаренко.— Харьков, 2014.— С. 22–36.*

6. ***Орлов, С. А.*** *Технологии разработки программного обеспечения: учебник для вузов.— 4-е изд. Стандарт третьего поколения / С. А. Орлов, Б. Я. Цилькер. — СПб.: Питер, 2012.— 608 с.*

7. ***Доренський, О. П.*** *Синтез структури комплексної моделі об'єктно-орієнтованого програмного забезпечення / О. П. Доренський // Системи обробки інформації. — 2014.— Т. 2, вип. 2(118).— С. 68–72.*

8. ***Макгрегор, Д.*** *Тестирование объектно-ориентированного программного обеспечения: практ. пособие.— Пер. с англ./ Джон Макгрегор, Девид Сайкс.— К.: ООО «ТИД ДС», 2002.— 432 с.*

9. ***Гилл, А.*** *Введение в теорию конечных автоматов / Артур Гилл.— М.: Наука, 1966.— 272 с.*

10. ***Сигорский, В. П.*** *Математический аппарат инженера / В. П. Сигорский.— Харьков: Техника, 1975.— 768 с.*

11. ***Глушков, В. М.*** *Синтез цифровых автоматов / В. М. Глушков.— М.: Физматгиз, 1962.— 476 с.*

12. ***Кузнецов, О. П.*** *Дискретная математика для инженера / О. П. Кузнецов, Г. М. Адельсон-Вельский.— М.: Энергоатомиздат, 1988.— 480 с.*

13. ***Доренський, О. П.*** *Перетворення моделі станів екземплярів класу об'єктів програмного забезпечення до автомата Мура / О. П. Доренський // Інженерія програмного забезпечення 2014: міжнар. наук.-практ. конф.: тези доп.— К.: НАУ, 2014.— С. 25–26.*

14. ***Доренський, О. П.*** *Особливості представлення структури компонент інтегрованої моделі об'єктно-орієнтованого програмного забезпечення інформаційних систем / О. П. Доренський // Системи обробки інформації. Проблеми та перспективи розвитку IT-індустрії: VI Міжнар. наук.-практ. конф.: тези доп.— 2014.— Т. 2, вип. 2(118).— С. 239.*

А. П. Доренский, А. А. Смирнов

ФОРМАЛИЗАЦИЯ ПРОЦЕССА ИЗМЕНЕНИЯ СОСТОЯНИЯ ПРОГРАММНЫХ ОБЪЕКТОВ СЛОЖНЫХ СИСТЕМ НА ОСНОВЕ КОНЕЧНЫХ АВТОМАТОВ МУРА

В предлагаемой статье модель состояний представителей класса объектов программного обеспечения сложных систем, разрабатываемого на основе объектно-ориентированного подхода, рассмотрена с точки зрения абстрактного синтеза конечного автомата. Обоснован и использован инициальный конечный автомат Мура. Осуществлено преобразование модели состояний в конечный автомат. Раскрыта сущность формализации последовательности событий полученной конечно-автоматной модели поведения представителей класса объектно-ориентированного программного обеспечения, а также особенности формализации результатов ее функционирования на стадии проектирования программных систем.

Ключевые слова: программное обеспечение; объектно-ориентированный подход; проектирование; программный объект; состояние; поведение; модель; конечный автомат Мура.

O. P. Dorensky, A. A. Smirnov

PROCESS STATE CHANGES FORMALIZATION OF SOFTWARE OBJECTS IN COMPLEX SYSTEMS BASED ON MOORE MACHINES

In this paper states model class instances in complex software systems, developed basing on object-oriented approach, is considered as abstract synthesis of a finite-state machine, proved and used Moore machine. Transformation model in a finite-state machine was done, presented the essence of formalization events sequence of result model of behavior of object-oriented software class instances, presented features of formalizing the results of theoperation model at the software systems design stage.

Keywords: software; object-oriented approach; design; program object; state; model; Moore machines.