УДК 004.896:681.3:621.311

І.А. КОТОВ*, канд. техн. наук, доц.
Криворізький національний університет

АЛГОРИТМІЗАЦІЯ МЕРЕЖІ СТАНІВ ТРИГЕРНОЇ ІНТЕЛЕКТУАЛЬНОЇ ПІДСИСТЕМИ ДИСПЕТЧЕРИЗАЦІЇ ЕНЕРГОСИСТЕМИ ГІРНИЧО-МЕТАЛУРГІЙНОГО КОМПЛЕКСУ

Метою статті є виклад результатів розробки принципової укрупненої тригерної схеми алгоритму функціонування системи підтримки прийняття рішень (СППР). Запропонована модель візуалізації алгоритмів у вигляді тригерної мережі станів, що має взаємодію із зовнішнім середовищем. Введена нова інтерпретація компонентів мережевої триггерной моделі. Модель тригерної системи підтримки рішень інтерактивно пов'язана як з діями користувача-оператора, так і зі станами компонентів енергосистеми. Проблема реалізації операцій управління формами знань, які являють собою приватні програмні контролери знань, ініційовані тригерами, є актуальною.

Методи дослідження полягають в застосуванні автоматної моделі станів та тригерної моделі функціонування програмної системи підтримки рішень. При цьому стан автоматної моделі пов'язується з виконанням пакета метаправил для управління логічним висновком. У роботі використані методи теорії множин, математичної логіки, теорії автоматів, електроенергетичних систем (ЕЕС), теорії графів, математичної статистики.

Наукова новизна полягає в новій моделі подієвої взаємодії системи підтримки рішень з енергооб'єктами електроенергетичної системи. Для схеми станів реалізована автоматна модель функціонування з конкретизацією семантики станів, транзакцій і тригерів. Показано, що структурна схема тригерних станів реалізує базовий набір структур алгоритму. Розроблена схема візуалізації виконання транзакцій, як одного стану програмної системи. Здійснено узагальнення в єдину структуру еволюційної ієрархії всіх контролерів управління онтологіями знань.

Практична значимість роботи полягає в розробці алгоритмів синтезу баз знань і подієво-керованого логічного висновку, що дає можливість розробки надійних уніфікованих інтелектуальних систем високої складності, взаємодіючих з зовнішнім середовищем. Реалізація та впровадження програмного комплексу дозволить підвищити якість диспетчерського управління режимами і технологічними процесами електричних мереж.

Результати роботи. Розроблено новий формалізм подання алгоритмів керування базами знань, взаємодіючих з зовнішнім середовищем, котрий агрегує примітиви станів, тригерів і транзакцій операцій і узагальнює стандартні мови візуалізації алгоритмів. Це дозволяє описувати алгоритми роботи баз знань і подієво-керованого висновку, що забезпечує розробку надійних уніфікованих інтелектуальних систем, взаємодіючих з зовнішнім середовищем.

Ключові слова: транзакція, енергосистема, автомат, онтологія, евристика, база знань, інкорпорація.

doi: 10.31721/2306-5435-2019-1-106-24-30

Проблема і її зв'язок з науковими і практичними завданнями. Моделі використання різних форм подання знань в системі підтримки рішень реалізуються через модель узагальнених онтологій і управляються метаправилами системи. Робота метаправил принципово відрізняється від функціонування форм знань інших рівнів: метаправила грають роль керуючого інтерфейсу між подіями зовнішнього середовища і базою знань. Проблема, що розглядається в роботі, полягає в необхідності реалізації операцій управління конкретними формами знань, які являють собою приватні програмні контролери рівнів знань, ініційовані тригерами. Такий підхід забезпечує роботу інтелектуальної системи, як засобу автоматизації прийняття рішень і управління, а також дозволяє здійснювати перехід до роботи СППР в автоматичному режимі.

Входами тригерів є стані елементів зовнішнього середовища, що представляють собою безперервні або дискретні фізичні величини різних типів. Ці величини повинні оцифровуватися і уніфікуватися до єдиного формату, на який налаштовані тригери.

Таким чином, є актуальною проблема розробки процедур обробки знань, які знаходяться в еволюційних відносинах між собою. Контролери верхніх рівнів знань використовують роботу контролерів попередніх рівнів. Тригери, як сигнали управління процедурами (контролерами) обробки знань, з одного боку, повинні представлятися концептами в просторі бази знань, а, з іншого боку, повинні забезпечувати зв'язок із зовнішнім середовищем.

Аналіз досліджень і публікацій. Визначальними якостями функціонування проектованої СППР є складність і об'ємність об'єкта управління − енергосистеми гірничо-металургійного комплексу, зв'язок із засобами ОІУК АСДУ, необхідність роботи як в режимі реального часу, так і інтерактивно − в ролі порадника оперативно-диспетчерського персоналу, робота з різними формами представлення знань, поєднання обчислювальних задач і логічного висновку та інші.

Перераховані якості формують ряд особливих вимог як до самої інтелектуальної системи, так і до методології її проектування.

Вимоги обумовлюють необхідність нового ексклюзивного підходу до подання і візуалізації структурних і функціональних моделей проектованого програмного комплексу. В роботі проведено короткий аналіз існуючих функціональних моделей візуалізації алгоритмів (сценаріїв, поведінки), які найбільш адекватно реалізують поставлені в роботі завдання [1, 2]. Кожен з формалізмів має сильні сторони і деякі обмеження, а також переважну область застосування. Результати аналізу існуючих формалізмів функціональних моделей наведені в табл. 1.

Таблиця 1

Результати аналізу існуючих формалізмів функціональних моделей

Модель візуалізації	Результати аналізу
Текстуальна форма	Підхід застосовний, в основному, в колективах розробників, вже знайомих з попередніми варіантами вихідного коду. Інакше, вивчення чужого вихідного коду зажадає занадто багато часу
Блок-схеми алгоритмів	У міру зростання складності та ієрархічності завдання, схема алгоритму втрачає логіку в кількості зв'язків і переходів [1 − 3]
Діаграми Дейкстри	Мають обмеження сучасних блок-схем алгоритмів [4, 5]
Метод Нессі-Шнейдермана	Збільшується кількість кордонів блоків. Зменшення розмірів схеми зменшує внутрішні відділи, ніж знижується їх читабельність [6, 7]
Моделі автоматів Мілі і Мура	Відсутність ієрархії станів, узагальнення переходів, засобів відображення відновлення нормальної роботи після станів переривання, подання семантики взаємодії кінцевих автоматів [8, 9]
SWITCH-технологія	Необхідність підтримки великих переліків подій, вхідних і вихідних впливів, представлених в кінцевих програмних модулях у вигляді формальних ідентифікаторів (буква + номер), що зводяться, в кінцевому рахунку, до числових кодів (номерів), необхідність ретельного протоколювання глобального списку подій [10, 11]
Диаграми станів Д. Харела	Відсутня можливість візуалізувати вплив подій зовнішнього середовища на умови запуску станів [12, 13]
Р-схеми	Концентрація на поданні потоку управління і відсутність відображення потоків даних. Також складно представляються кілька точок входу в модель [14, 15].
Мова ДРАКОН	Велика кількість графічних примітивів (ікон − 25 штук, макроікон − 20 штук), дві моделі представлення завдань, відсутність явного подання подій і взаємодії із зовнішнім середовищем [16, 17]
L-мережі	Теоретичне обгрунтування, лаконічна формальна мова, однакове подання «малих» і «великих» функціональних структур [18, 19]

Узагальнюючи розглянуті вище основні підходи до моделювання та візуалізації процесів проектування складних програмних систем, слід відзначити їх безсумнівну корисність в певних областях і умовах. Однак, ґрунтуючись на проведеному аналізі, можна сказати, що вони не цілком відповідають завданню проектування подієвої інтелектуальної СППР, що активізуються зовнішнім середовищем. Тому сформульовані вище вимоги до засобів візуалізації моделі структурно-інформаційної схеми програмного комплексу СППР обумовлюють актуальність і необхідність розробки нових синтетичних засобів подання структури і функціонування СППР, яка взаємодіє з комплексом АСДУ енергосистеми і оперативним персоналом.

Постановка завдання. Проектування структури і методів роботи програмного комплексу системи підтримки рішень вимагає врахування особливостей реалізованих нею алгоритмів, операцій, властивостей і механізмів функціонування бази знань на основі еволюції інкорпорації професійних онтологій, а також специфіки середовища виконання програми. Деякими визначальними якостями функціонування проектованої СППР є: складність і об'ємність об'єкта управління − енергосистеми гірничо-металургійного комплексу, зв'язок із засобами ОІУК АСДУ, необхідність роботи як в режимі реального часу, так і інтерактивно − в ролі порадника оперативно-диспетчерського персоналу, робота з різними формами подання знань, поєднання обчислювальних задач і логічного висновку та інші.

Перераховані якості формують ряд особливих вимог як до самої інтелектуальної системі, так і до методології її проектування. Наведемо базові з таких вимог:

одночасна робота з декларативними і процедурними формами подання знань;

однакова робота системи в режимі логічного висновку і обробки бази даних;

забезпечення апаратного інтерфейсу з компонентами енергооб'єктів та АСДУ;

реалізація призначеного для користувача візуального інтерфейсу з ОДП;

ефективна трансляція у внутрішні коди системи і інтерпретація керуючих метаправил;

реалізація програмного механізму логічного висновку у вигляді транзакцій метаправил.

Викладення матеріалу та результати. Розробимо модель мережі станів. Один стан є блок з тригера, операційної транзакції, результату обчислень і вихідних символів (кодів). Конкретизуємо ці компоненти.

Один стан моделі функціонування системи підтримки рішень (q) – це логічно виділений етап її функціонування, пов'язаний з реалізацією однієї конкретної транзакції.

Тригер – це зовнішній апаратний або програмний об'єкт, що генерує коди, які розпізнаються СППР як вхідний алфавіт (V) і переводять її в один, однозначно пов'язаний з даними тригером, стан (q).

Зі станом однозначно пов'язаний функціональний блок транзакції. Транзакція (p) – это обчислювальний блок, який реалізує послідовний набір команд, виразів, операцій і розглянутий в рамках поточної транзакції як один обчислювальний пакет. Кожна транзакція жорстко пов'язана з конкретним станом (q). Транзакції можуть містити набори метаправил для управління БЗ і виробництва логічного висновку, пакети запитів для управління БД, оператори математичних обчислень, команди управління поведінкою СППР і т.д. В результаті виконання транзакції формується результат транзакції (r) і коди з вихідного алфавіту системи (W).

Введемо інтерпретацію компонентів триггерной мережі станів СППР. Приймемо, що вертикальна риса позначає стан мережі (на відміну від переходу в мережах Петрі), а дуга − перехід (на відміну від вертикальної риси в мережах Петрі).

Множина операційних позицій (станів) мережі

$$S =< Q, q_0, F, T, P >,$$
$$S =< s_i \mid s_i =< q_i, t_i, p_i >, q_i \in Q \vee q_i \in F \vee q_i = q_0 > . \quad (1)$$

Множина переходів мережі

$$\Delta_\Sigma = \Delta \bigcup \Delta_t = \{\delta_i \mid \delta_i \in \Delta \vee \delta_i \in \Delta_t\}, \quad (2)$$

де Δ_Σ − загальна множина переходів мережі; Δ − множина переходів між станами мережі; Δ_t − множина тригерних переходів мережі.

Множина функцій входу

$$I(S): S \to < Q_{-1}, V_\Sigma \bigcup W_\Sigma >,$$
$$I(s_i): s_i \to \{q_{i-1}, v_{i-1} \bigcup w_{i-1}\}, \quad (3)$$

де Q_{-1} − множина попередніх станів по відношенню до поточного стану.

Множина функцій виходу

$$a: S \to s_a, \quad (4)$$

Таким чином, звернемо автоматну тригерну модель в операційну тригерну мережу станів, узагальнена модель якої тепер може бути записана в наступному вигляді

$$N_S =< S, \Delta_\Sigma, I(S), O(S) > . \quad (5)$$

У найбільш загальному вигляді приклад моделі (5) можна представити наступною графічною інтерпретацією, наведеною на рис. 1.

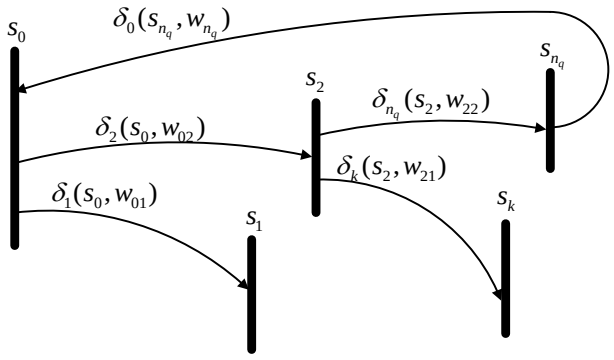

Рис. 1. Приклад узагальненої моделі станів програмної системи

У запропонованій моделі, як випливає з (5), стан являє собою блок з тригера, операційної транзакції, результату обчислень і вихідних символів (кодів). Стан будемо відображати вертикальної рисою з ідентифікатором стану вгорі. Залежно від отриманих вихідних кодів відбувається розгалуження процесу роботи мережі – перехід до наступного стану. Якщо стан має більше одного виходу, то вибирається строго один перехід. Виконання даної умови гарантує однозначність і детермінованість виконуваного алгоритму. Тактом роботи мережі є активність одного поточного стану. Таким чином, в кожному такті роботи однієї мережі може бути активним тільки один

стан. Для вирішення питання моделювання паралелізму програмних процесів задається єдиний тригер для запуску декількох паралельних операційних мереж станів.

Наведені міркування дозволяють ввести поняття і функцію маркування для розробленої моделі операційної мережі станів. Функція маркування, по суті, є функцією активізації і реалізує вибір єдиного активного стану зі всієї безлічі станів мережі

$$a : S \to s_a, \tag{6}$$

де $-\ \#(s_a)=1 | s_a \in S$ єдиний активний стан в даному такті роботи. Всі інші стани не активні − $\#(s_i) = 0 \mid \forall s_i \in S \wedge s_i \neq s_a$.

Маркована мережа станів матиме вигляд

$$N_S =< S, \Delta_\Sigma, I(S), O(S), a > . \tag{7}$$

На основі розробленої моделі стану можна сформулювати загальний принцип виконання мережі. Виконання мережі (хід реалізації алгоритму обчислень програмної системи) визначається переходами між станами або активізацією станів, що означає потактову перемаркировку уздовж деякого шляху на графі станів.

Активізація поточного стану відбувається при спрацьовуванні тригера з викидом вихідного сигналу (символу) − w, або при отриманні вихідного сигналу (символу) попереднього стану. При цьому, в стані генерується символ запуску транзакції − ν_t. Якщо символ запуску розпізнано транзакцією, то вона виконується, в іншому випадку − процес зупиняється (або виконується операція зупинки). При виконанні внутрішніх умов в транзакції, або при її завершенні відбувається перехід до наступного стану по одній з дуг розгалуження відповідно до функції переходу − δ. Поточний стан зберігає результат виконання транзакції − r.

Рішення завдання на мережі станів програмної системи є реалізацією алгоритму, вираженого шляхом на графі переходів. Згідно з теоремою про структурування, схема алгоритму може бути зведена до базисного набору структур: послідовний хід, розгалуження і повернення (цикли) [20]. Тригерна модель станів дозволяє реалізувати ці принципи.

Приклад виконання деякого алгоритму уздовж шляху станів $s_0 \to s_2 \to s_{nq} \to s_0$ розглянутої мережі відобразиться наступними тактами, наведеними на рис. 2. Запуск виконання провадиться за тригером t_0.

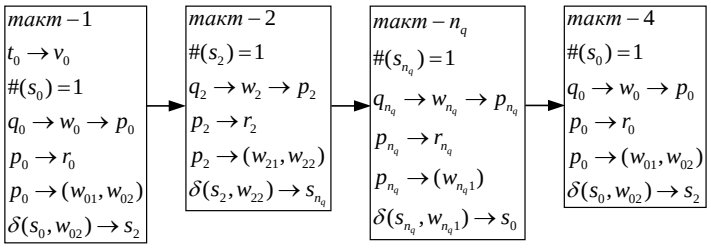

Рис. 2. Приклад кадрів потактового виконання алгоритму на триггерной мережі станів

На основі запропонованих моделей (1) – (7) деталізуємо модель одного стану, яка показана на рис. 3.

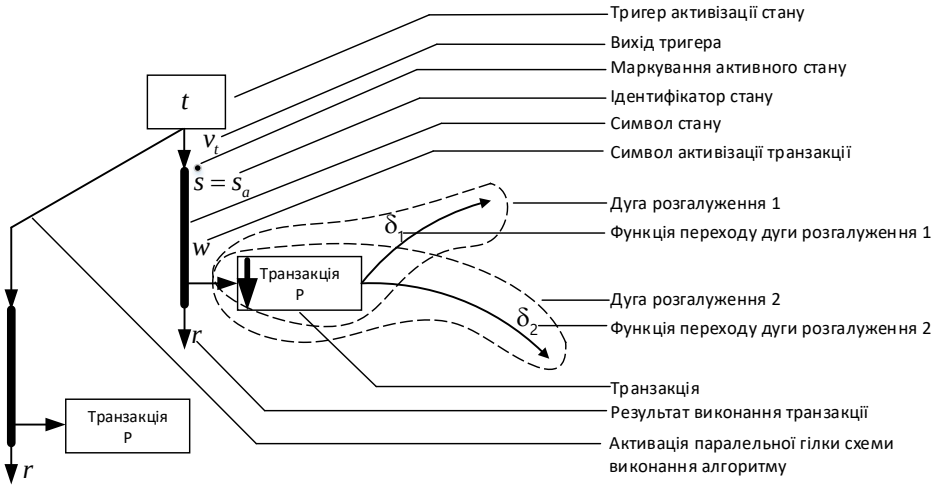

Рис. 3. Графічна інтерпретація одного стану виконання алгоритму програмної системи

Тепер деталізуємо наведену вище графічну інтерпретацію одного стану виконання алгоритму транзакції програмної системи з урахуванням введених моделей і покажемо її на рис. 4. На рисунку показані всі складові моделі − стан, тригер, транзакція і операція.

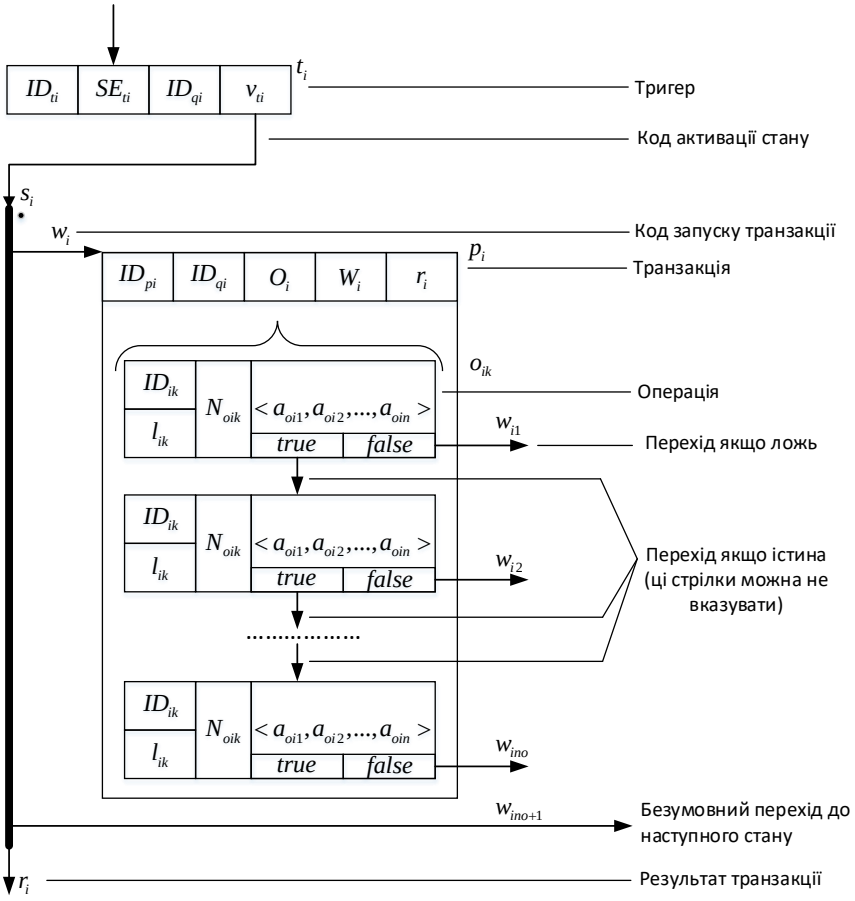

Рис. 4. Загальна схема візуалізації виконання транзакції одного стану програмної системи

Реалізація запропонованої моделі візуалізації вимагає розробки формальної мови схеми формування гілок алгоритму функціонування триггерной програмної машини станів

$$L(G)^T =< N, \Sigma, S, P >, \tag{8}$$

де N – алфавіт нетерміналів; Σ – алфавіт терміналів, $N \cap \Sigma = \varnothing$; S − стартовий символ граматики G; $a \in N$, $b \in (N \cup \Sigma)^*$: $a \to b$ – правила виведення граматики G.

Врахуємо, що програма являє собою підмножину множини шляхів на графі мережі станів, як було показано на прикладі (рис. 1)

$$\begin{aligned} &PROG = B_{prog}, \\ &B_{prog} = \{b_i \mid i = 1, n_b\}, \\ &b_i = \{s_{i,k}, s_{i,k+1}, ..., s_{i,nbi}\}, \\ &\forall s_{ik} \forall s_{ik+1} \forall k \mid s_{ik} \in S \wedge s_{ik+1} \in S_i \wedge k = 1,2,..., n_0 : s_{ik} \prec s_{ik+1}, \\ &B_{prog} \subseteq N_S, \end{aligned} \tag{9}$$

де $PROG = B_{prog}$ − програма (алгоритм виконання серії транзакцій); $B_{prog}=\{b_i|i=1,n_b\}$ − множина гілок (шляхів) алгоритму виконання програми; b_i − і-а гілка виконання алгоритму програми, задана як строга послідовність станів тригерної мережі; N_s − повна мережа станів.

Таким чином, отримана модель візуалізації алгоритмів, представлена як тригерна мережа станів програмної системи, що має взаємодію із зовнішнім середовищем. Операції в моделі можна розглядати на різних рівнях.

Якщо операції розглядати як елементарні команди формату

<код операції><коди операндів>,

то модель дозволяє відображати роботу автоматного процесора або інтерпретатора на низькому рівні машинних команд. Разом с тим, модель дозволяє візуалізувати структури алгоритмів на мовах програмування високого рівня.

Висновки та напрямок подальших досліджень. Розроблені структура і принципова укрупнена тригерна схема алгоритму функціонування СППР. Запропонована модель візуалізації алгоритмів, як тригерна мережа станів програмної системи, що має взаємодію із зовнішнім середовищем. Введена нова інтерпретація компонентів тригерної моделі.

Напрямок подальших досліджень полягає в розробці схеми інтеграції БЗ і блока логічного висновку, які функціонально перекриваються, на відміну від класичної архітектури інтелектуальних систем. Передбачається додатково включити в структуру СППР селектор рівня онтологій для оперативного вибору форми подання знань і даних в режимі реального часу, а також модулі динамічного формування, верифікації та поповнення онтологій. Планується подальша розробка інформаційних та функціональних трактів взаємодії програмних тригерів з апаратно-програмними засобами ОІУК АСДУ ЕЕС.

Список літератури

1. **Пышкин Е.В.** Структуры данных и алгоритмы: реализация на C/C++ / Е.В. Пышкин − СПб.: ФТК СПБГПУ – 2009 − 200 с.
2. **Wirth N.** Algorithms + Data Structures = Programs. Prentice-Hall, Inc. Englewood Cliffs, N.J. 1976.
3. **Brooks, Frederick P., Jr**. The Mythical Man-Month. Essays on Software Engineering. Anniversary Edition. Addison − Wesley, 1995.
4. **Dahl O.-J., Dijkstra E.W., Hoare C.A.R.** Structured Programming. Academic Press. London and New York. 1972.
5. **Дал У., Дейкстра Э., Хоор К.** Структурное программирование − М.: Мир – 1975 − С. 25–28
6. **Nassi I, Schneiderman B.** Flowcharts techniques for structured programming // ACM SIGPLAN Notices, Vol.8, No.8, August 1973 − pp. 12 – 26
7. **Cornelia M.Y., Marilyn L.S.** Nassi-Shneiderman charts an alternative to flowcharts for design − ACM SIGSOFT/BIGMETRICS Software and Assurance Workshop, November 1978 – pp. 386 − 393
8. **Поликарпова Н.И., Шалыто А.А.** Автоматное программирование – СПб.: Изд-во СПбУ − 2008 − 167 с.
9. **Хопкрофт Д., Мотвани Р., Ульман Д.** Введение в теорию автоматов, языков и вычислений − М.: Вильямс − 2002 − 528 с.
10. **Шалыто А.А., Туккель Н.И.** SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // «Программирование», №5 – 2001− с. 45 − 62
11. **Шалыто А.А.** SWITCH-технология. Алгоритмизация и программирование задач логического управления. - СПб.: Наука, 1998.- 628 с
12. **Harel D.** Statecharts: A visual formalism for complex systems // Science of Computer Programming. June 1987 − Vol. 8, No. 3 − pp. 231 – 274
13. **Harel D.** On Visual Formalisms // Communications of the ACM − Vol.31, No.5, May 1988 − pp.514 − 530
14. **Вельбицкий И.В.** Безбумажная Р-технология проектирования широкого применения// Выч. техника социалистических стран, вып.18 − М.: Финансы и статистика, 1985.- с. 21-39
15. ГОСТ 19.005-85. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения. – Действует с 01.07.1986.
16. **Титова Е.В.** Алгоритмический язык Дракон в лингвистике // Сборник работ 68-й научной конференции студентов и аспирантов Белорусского государственного университета в трех частях. Часть 3// Минск: БГУ – 2011 − С. 50 − 52
17. **Калиногорский Н.А.** Автоматизация процесса разработки алгоритмов управления в интегрированной среде Дракон 2007-2010 / Сост. Н.А. Калиногорский − Новокузнецк: Изд. центр СибГИУ − 2013 − 50 с.
18. **Lekarev M.F.** Das graphische Verfahren der Software-Entwicklung für logisch komplizierte Anwendungen // Technische Berichte der Fachhochschule Hamburg, №25 (Aug.1993), s.36-38.
19. **Лекарев М.Ф.** Организация программного обеспечения и вычислительных процессов ЭВМ на базе визуального формализма: дис. … д-ра техн. наук: 05.13.13, 05.13.11 – СПбГТУ, Санкт-Петербург, 1998 – 238 с.
20. **Linger, Richard S. Mills, Harland D. Witt, Bernard I.** Structured Programming: Theory and Practice. Addison-Wesley, 1979.

Рукопис подано до редакції 26.09.2019