

С.Л. Кривий, О.О. Гріненко

МОДЕЛЬ ЕКОСИСТЕМИ “КАФЕДРА”

Пропонується модель екосистеми, в якій описані основні об'єкти та їхні функції у такому середовищі вищого навчального закладу як кафедра. Основним об'єктом дослідження в такій моделі є умови успішного розвитку та функціонування кафедри. Зокрема розглядаються способи планування рівномірного навчального навантаження на викладачів, взаємодія працівників кафедри між собою і керівництвом факультету. Основними діючими особами в екосистемі є викладачі (названі серверами), які викладають відповідні курси. Пропонована модель дозволяє спростити складання розкладу, оперативно реагувати на форс-мажорні обставини, необхідні заміни викладача тощо. Властивості моделі екосистеми досліджуються автоматом-мережевими методами. Така екосистема може узагальнюватися та розширюватися шляхом надбудови моделі факультету, університету, міністерства з метою контролю роботи факультету, університету, міністерства та ефективності їхнього функціонування.

Ключові слова: модель, екосистема, властивості.

S.L. Kryviy, O.O. Grinenko

MODEL OF THE “DEPARTMENT” ECOSYSTEM

An ecosystem model is proposed, which describes the main objects and their functions in such medium of a higher educational institution as a department. The main object of research in such models are conditions of the successful development and functioning of the department. In particular, the methods of planning an equal educational load for teachers, together with the employees of the university department and the management of the faculty are considered. The main actors in the ecosystem are teachers (called servers), who teach the courses. The proposed model makes it possible to simplify the drawing up of the schedule, to quickly respond to force majeure circumstances, necessary exchange of the teacher, etc. The properties of ecosystem models are verified by automata-network methods. Such ecosystem can be generalized and expanded by adding models of the faculty, university, and Ministry in order to control the work of the faculty, university, and ministry, the effectiveness of their functioning.

Key words: model, ecosystem, properties.

Вступ

В [1] була запропонована модель екосистеми програмного забезпечення (ПЗ), яка в даній роботі адаптується до побудови моделі екосистеми, що обмежується областю “Кафедра” як основної ланки функціонування закладу вищої освіти (ЗВО). На жаль, обмеження на обсяг статті не дозволяють описати детально всі властивості моделі, тому описуються лише основні етапи її функціонування протягом одного робочого дня.

Загальна модель екосистеми

Об'єктом дослідження в екосистемі, яка належить до навчання у ЗВО, є методи, моделі і засоби, які дають можливість досліджувати взаємодію об'єктів з оточуючим середовищем, та взаємодію об'єктів екосистеми між собою. Водночас головним

аспектом такої взаємодії є стійкий розвиток екосистеми в цілому.

Предметом дослідження є процеси, які відбуваються на такому об'єкті ЗВО як кафедра та її оточенні (методи, взаємодія, викладачі, студенти, розробники, замовники, контролюючі органи тощо). Далі будемо користуватися таким загальним означенням екосистеми [1].

Означення 1. Екосистемою (ЕС) називається четвірка $ES = (EE, E, B, SEB)$

- EE – зовнішнє середовище;
- E – автономне сімейство об'єктів, які взаємодіють між собою і з зовнішнім середовищем (ядро ЕС);
- B – база знань, де зберігається еволюція розвитку сімейства E як популяції (макрознання ЕС);

- S_{EB} – сервер для забезпечення взаємодії між E , B та зовнішнім середовищем.

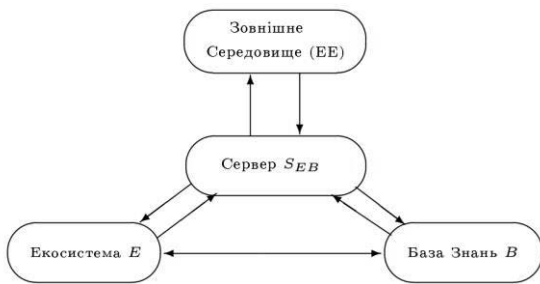


Рис. 1. Модель екосистеми верхнього рівня

Зовнішнє середовище EE – це зовнішнє оточення, в якому функціонує екосистема (кафедра). До цього об'єкта належить усе, що має зовнішній вплив на це функціонування.

Сімейство об'єктів E в EC загалом можна інтерпретувати по-різному, але в даному розгляді ці об'єкти отримують цілком конкретну інтерпретацію, про яку йтиме мова далі.

База знань B в EC відіграє роль сховища знань про етапи розвитку кафедри, про розроблені курси, лабораторні роботи та практичні завдання, виконані проекти, інформацію про існуюче та нове ПЗ, відомості про допущені помилки у виборі курсів і завдань, про допущені помилки керівництва, позитивний досвід, ухвалені рішення, про ПЗ, яке використовується кафедрою, про досвід і стаж роботи викладачів, які працюють в даний час і про тих, які залишили кафедру, про структуру кафедри, освітні програми, навчальні програми, програми окремих курсів, про завдання, які виконані, і які виконуються працівниками кафедри тощо.

Сервер S_{EB} служить для того, щоб забезпечувати оперативну взаємодію між E , B та зовнішнім середовищем EE . Система E може безпосередньо працювати із БЗ у разі, коли немає потреби використовувати сервер S_{EB} . А коли потрібно отримати доступ до певної приватної чи конфіденційної інформації, то необхідно дістати дозвіл на доступ до такої інформації. Цю функцію, зокрема, виконує сервер S_{EB} . Крім того, у

даній екосистемі основними діючими особами такого сервера є завідувач кафедри, професори, гаранті освітніх програм, адміністратор системи. До їхніх функцій належать комунікація з зовнішнім середовищем (аналіз ринку, наукових досягнень, взаємодія з деканатом, ректоратом та з колегами інших кафедр, факультетів та вузів).

Складові ЕС та їхня характеристика

Сімейство E , як ядро ЕС, зображується такою четвіркою, яку будемо називати **серверною частиною ЕС**,

$$E = (S, \Omega, o, f) \quad (1)$$

де

– $S = \{S_1, \dots, S_n\}$ – скінченна множина, елементи якої називаються серверами, в нашому випадку - це викладачі кафедри;

– $Q = \{w_1, \dots, w_n\}$ – скінченна множина операцій, які можуть виконуватися різними серверами, в нашому випадку – це курси, які можуть викладатися викладачами кафедри;

– $o: B \rightarrow \Omega$ – множина операцій, які може виконувати окремо взятий сервер, в нашому випадку - це курси, які може читати і читає конкретний викладач кафедри;

– $f: S \times \Omega \rightarrow D \times Q^+$ – функція, значеннями якої є рік, місяць, день (область D) і час виконання окремої операції на окремо взятому сервері в нашому випадку – це конкретний курс, який читає або читав конкретний викладач в певний період часу.

База Знань. Створення БЗ ґрунтується на використанні мов дескриптивних логік і передбачає означення множини атомарних концептів та атомарних ролей, створення термінології ($TBox$), створення множини фактів ($ABox$) та вибір логічної мови і алгоритмів для цієї мови, за допомогою яких відбувається генерація відповідей на запити до БЗ. В термінах $TBox$ і $ABox$

описується еволюція EC у часі, етапи розвитку, крах кафедри тощо. Базовим ядром дескриптивних логік є логіка ALC , для алгоритмів якої існують ефективні реалізації [3, 4].

Крім того, БЗ виконує також роль сховища різного роду інформації: теми дипломних робіт, дата захисту дипломної роботи, програмні продукти, розроблені в дипломних роботах, відомості про працевлаштування випускників кафедри, особові справи та дані викладачів, відомості про стаж, терміни стажування, зміни в становищі працівника кафедри, зміни адреси, номера телефону тощо. Ця інформація доступна лише особам уповноваженим, зокрема, адміністратору БЗ. Наявність такої БЗ дає можливість оперативно реагувати на запити надання інформації про ту чи іншу особу.

Сервер обміну S_{EB} забезпечує оперативну взаємодію між E, B та оточуючим середовищем. На цьому сервері знаходяться запити до БЗ, які ще не були виконані (в ситуації великого завантаження), відповіді на запити, які вже опрацьовані, але не були затребувані, та поточна інформація про стан БЗ і системи E . Основною функцією цього сервера є взаємодія з зовнішнім середовищем EC . Ця взаємодія полягає в отриманні інформації про зміни, які відбуваються у зовнішньому середовищі, стан та потреби ринку, досягнення конкурентів, нові тенденції в розробці ПЗ, його ринкова вартість тощо.

Серверна частина EC . В модель серверної частини вкладаються в даному випадку викладачі кафедри (сервери), які задіяні в E . Дійсно, множина S – це множина викладачів (штатні викладачі, сумісники), Q – це курси, які можуть викладатися і викладаються викладачами, $o(S_i) = \Omega_i$ – це сукупність курсів (предметів), які може викладати конкретний викладач S_i , $i = 1, \dots, |\Omega|$, $f(S_i, w_j) = (y, m, d, t_i)$, де $S_i \in S$, $w_j \in \Omega$, $(y, m, d) \in D$, $t_i \in Q^+$ – час виконання викладачем S_i операції w_j . Водночас

сервери можуть передавати управління один одному (в нашому випадку передача управління робочими серверами один одному регламентується розкладом), де такими серверами є асистенти, доценти, професори. У разі форс-мажорних ситуацій (неявка студентів, відсутність освітлення, відсутність умов проведення заняття, хвороба викладача тощо) передбачається взаємозв'язок з завідувачем кафедри, методистами кафедри, деканатом, заступниками декана, деканом і передача їм управління для розв'язання проблеми.

Значення функції f відіграють важливу роль у EC , за допомогою якої проводиться обчислення навантаження на викладачів та контроль за виконанням завдань. Ці значення – кількісні (часові) оцінки складності виконання операцій в найгіршому випадку, або взяті з деяких емпіричних спостережень. У даному випадку враховується як кількість та час виконання навантаження, так і час, витрачений на підготовчі дії (створення курсу лекцій, розробка практичних, семінарських занять та лабораторних робіт.), чим і визначається вибір викладачів для реалізації відповідних операцій. Маючи в розпорядженні значення функції f , можна прогнозувати ризики в EC з урахуванням стажу, досвіду та кваліфікації викладачів (ця інформація береться з БЗ B). Прогнозування затримок у роботі або ризиків зриву виконання планових занять, передбачених розкладом, дає можливість швидко реагувати на виклики в EC . Цього можна досягти шляхом вчасних замін, завчасного інформування студентів про зміни в розкладі та перенесення занять (за погодженням із деканатом).

Крім того, значення функції f дають підставу для пошуку оптимального шляху виконання завдань в EC , оскільки в моделі (1) припускається, що одне і те саме завдання може виконуватися різними серверами. А це дає можливість використовувати оптимізаційні алгоритми для побудови такого плану проведення занять, який

рівномірно розподіляє навантаження на викладачів, передбачене нормативами, та кафедральне обладнання.

Дослідження властивостей моделі

Розглянемо деякі підходи до дослідження властивостей моделі (1) та ілюстрацію введеного формалізму на прикладі системи E . В загальному випадку така система може мати велику сукупність об'єктів, і тоді виникає потреба у оперативному плануванні й не просто плануванні, а оптимальному плануванні й управлінні виконанням завдань та доступом до її спільних ресурсів. Можна запропонувати декілька методів побудови оптимального планування та управління. Складність розв'язання цих задач полягає в тому, що в загальному випадку ці задачі несуть у собі суперечності об'єктивного характеру. Це і є причиною існування декількох методів, оскільки для успішного розв'язання такого типу задач вибирається певний рівень абстракції, для якого і розробляються відповідні методи. Такого типу проблеми частіше за все формулюються у вигляді деякої оптимізаційної задачі, а потім розробляється метод розв'язання такої задачі.

Далі розглядається автоматно-мережева модель розв'язання задач планування розподілу базових ресурсів у системі E [5, 6].

Нехай в системі (1)

$$\Omega_i = \{w_1^i, w_2^i, \dots, w_{|\Omega_i|}^i\} \quad (2)$$

означає множину курсів (операцій), які можуть викладатися (виконуватися) i -м викладачем (сервером): $i = 1, 2, \dots, |S|$, $w_1^i, w_2^i, \dots, w_{|\Omega_i|}^i \in \Omega$. Отже,

$$o(S_i) = \Omega_i \quad (3)$$

З рівностей (2) і (3) отримуємо, що

$$f(S_i, w_k^i) = (y, m, d, t_k^i), \quad (4)$$

$$i = 1, 2, \dots, |S|, k = 1, 2, \dots, |\Omega_i|$$

Оскільки в системі допускається можливість викладання одного і того предмету різними викладачами, то це дає можливість рівномірного розподілу навантаження і часу для різних викладачів. Отже, $\Omega_i \cap \Omega_j \neq \emptyset$, ($i \neq j$).

Звідси випливає, що в такій постановці задача планування розкладу може мати декілька способів реалізації в системі E . Дійсно, довільне p -те завдання в E визначається розкладом занять і є послідовністю занять (операцій)

$$\overline{O}_p = o_{p_1} o_{p_2} \dots o_{p_k} \quad (5)$$

де $o_{p_1}, o_{p_2}, \dots, o_{p_k} \in \Omega$ курси, передбачені завданням. Оскільки кожне таке завдання визначається власною послідовністю типу (5) і існує декілька способів його реалізації викладачами, то існують і різні варіанти реалізації \overline{O}_p . В зв'язку з такою ситуацією накладемо таке обмеження:

Умова 1: початок виконання кожним викладачем наступного завдання (операції) неможливий доти, доки він не закінчить виконання попереднього завдання (операції).

Для того, щоб реалізувати це обмеження та оперативний зв'язок із керівництвом та технічними службами у випадку форс-мажорних ситуацій, виділимо у системі E спеціальні сервери X і Y , які будемо називати **вхідним** і **вихідним** відповідно, на відміну від інших серверів, які будемо називати **робочими**. Вхідний і вихідний сервери є підсистемами, які забезпечують взаємодію робочих серверів між собою і керівництвом факультету або іншим керівним органом, якому підпорядкована кафедра. За допомогою вхідного сервера керівництво викладає план реалізації завдання j -го дня (розклад-специфікацію), а за допомогою вихідного сервера дистанційно контролюється виконання завдання робочими серверами, виконуються оголошення, озвучуються накази, сповіщається про небезпеку (наприклад, про повітряну тривогу) тощо. Крім того, на вихідний сервер викладаються робочими серверами відомості про аварійне завершення виконання завдання з метою узгодження місця та часу проведення перерваного завдання з керівним органом (наприклад, деканатом). Звідси випливає, що вихідний сервер має певний пріоритет щодо вхідного і робочих серверів.

Отже, початок виконання деякої послідовності операцій в системі (1) може

специфікуватися в термінах вхідного та робочих серверів, а її закінчення – в термінах вихідного сервера. Взаємодія між серверами повинна синхронізуватися в зв'язку з тим, що різні операції можуть виконуватися декількома серверами. Така синхронізація передбачається розкладом-специфікацією виконання завдання. Але для оперативного реагування на непередбачувані випадки і забезпечення виконання умови 1, вводяться спільні булеві змінні $b_1, b_2, \dots, b_m = 0$, значення яких може читати кожний із серверів шляхом комунікації з вихідним сервером. Тоді, наприклад, значення $b_1 = 0, b_2 = 1, b_3 = 0$ означають, що сервер S_2 зайнятий, а сервери S_1 і S_3 вільні і готові виконувати операції завдання.

Модель екосистеми “Кафедра”

Для того, щоб продемонструвати підхід до аналізу властивостей наведеної моделі ЕС, розглянемо приклад дослідження властивостей моделі екосистеми “Кафедра”, яка має сім робочих серверів виконання завдання одного робочого дня кафедри, одним вхідним, одним вихідним серверами та одним сервером управління. Таку екосистему можна представити у графічному вигляді (див. рис. 2). Зауважимо, що така конкретизація не обмежує загальності моделі, оскільки цю модель можна адаптувати та деталізувати відповідно до правил, які регламентують роботу конкретного ЗВО. Позначення S_m/S_n означає, що реалізація операції може виконуватися або на сервері S_m , або на сервері S_n .

На вхідний сервер X завантажується розклад-специфікація завдання (j -го робочого дня кафедри) – послідовність операцій (предметів) завдання. Коли виконання закінчується, то його результат подається на вихідний сервер Y . Для кожного завдання і кожної пари приписується множина серверів із множини $S = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ і множина операцій із $\Omega = \{w_1^1 - w_1^4, w_2^1 - w_2^4, w_3^1 - w_3^4, w_4^1 - w_4^4\}$. Одна і та сама операція може бути приписана до різних завдань.

Сервер управління D реалізує функції деканату, в його завдання входять функції регуляції виконання операцій або завдань у випадку форс-мажорних ситуацій (неможливість виконання операцій, хвороба виконавців, військові обставини тощо), розв'язання спірних або конфліктних питань. Крім того, однією з функцій цього сервера є узгодження дати і часу переносу виконання завдань чи окремих операцій завдання. Ця функція досить важлива, оскільки передбачає успішне функціонування деканату як органу управління і комунікації з виконавцями.

Графічне зображення розкладу-специфікації

Припустимо, що розклад завдання j -го дня роботи кафедри для чотирьох курсів бакалаврату специфікується даними з табл. 1.

Таблиця 1

Розклад-специфікація j -го робочого дня кафедри для бакалаврату

j -й день	Завдання 1		Завдання 2		Завдання 3		Завдання 4	
	операція	сервер	операція	сервер	операція	сервер	операція	сервер
1	w_1^1	S_1/S_4	w_1^2	S_4/S_2	w_1^3	S_5/S_3	w_1^4	S_7/S_1
2	w_2^1	S_2/S_3	w_2^2	S_5/S_1	w_2^3	S_4/S_6	w_2^4	S_5/S_3
3	w_3^1	S_2/S_4	w_3^2	S_3/S_6	w_3^3	S_7/S_2	w_3^4	S_4/S_2
4	w_4^1	S_3/S_6	w_4^2	S_3/S_5	w_4^3	S_1/S_4	w_4^4	$S_1/S_6/S_7$

Тоді, повертаючись до позначень із (2), дістаємо:

$$\begin{aligned} \Omega_1 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_2 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_3 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_4 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_5 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_6 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\ \Omega_7 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \end{aligned} \quad (6)$$

Процедури реалізації плану задаються такими послідовностями:

$$\begin{aligned} X_{1j} &= \{w_1^1, w_2^1, w_3^1, w_4^1\} = \\ (S_1 \vee S_4)(S_2 \vee S_3)(S_2 \vee S_4)(S_6 \vee S_3) \\ X_{2j} &= \{w_1^2, w_2^2, w_3^2, w_4^2\} = \\ (S_2 \vee S_4)(S_1 \vee S_5)(S_3 \vee S_6)(S_3 \vee S_5) \\ X_{3j} &= \{w_1^3, w_2^3, w_3^3, w_4^3\} = \\ (S_5 \vee S_3)(S_4 \vee S_6)(S_2 \vee S_7)(S_1 \vee S_4) \\ X_{4j} &= \{w_1^4, w_2^4, w_3^4, w_4^4\} = \\ (S_1 \vee S_7)(S_3 \vee S_5)(S_2 \vee S_4)(S_1 \vee S_6 \\ &\quad \vee S_7) \end{aligned} \quad (7)$$

де X_{ij} означає i -те завдання j -го дня роботи, $i = 1, 2, \dots, 6$.

На підставі розкладу-специфікації виконання завдань, визначеного табл. 1, можна реалізувати шляхи виконання завдань. За цих обставин результати виконання завдань фіксуються на вихідному сервері Y , де у разі невиконання якоїсь операції інформація про це транзитом через усі проміжні робочі сервери (або іншим можливим способом) передається на вихідний сервер.

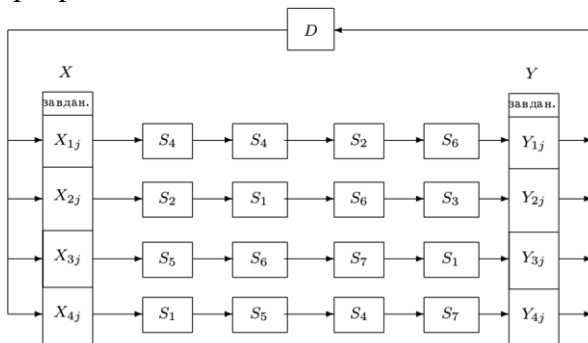


Рис. 2. Графічне зображення розкладу виконання завдань j -го робочого дня

Якщо на вихідному сервері фіксується невиконання якогось завдання або якоїсь операції такого завдання, то за погодженням із деканатом формується послідовність дій, яка передається на вхідний сервер для реалізації.

У разі успішного виконання завдання інформація про це теж передається на вихідний сервер, але така інформація не передбачає реагування.

Властивості моделі

Що корисного можна отримати з наведеної моделі, які властивості можна прослідкувати за її допомогою?

Перше, що впливає з наведеної моделі, – це можливість побудувати для викладачів зручний розклад реалізації завдань із врахуванням рівномірності розподілу навантаження на викладачів. Дійсно, виходячи із сказаного, отримуємо такі шляхи виконання першого завдання на підставі (7):

$S_1, S_2, S_2, S_6, S_4, S_2, S_2, S_6, S_1, S_3, S_2, S_6, S_4, S_3, S_2, S_6;$
 $S_1, S_2, S_2, S_3, S_4, S_2, S_2, S_3, S_1, S_3, S_2, S_3, S_4, S_3, S_2, S_3;$
 $S_1, S_2, S_4, S_6, S_4, S_2, S_4, S_6, S_1, S_3, S_4, S_6, S_4, S_3, S_4, S_6;$
 $S_1, S_2, S_4, S_3, S_4, S_2, S_2, S_3, S_1, S_3, S_4, S_3, S_4, S_3, S_4, S_3.$

Серед цих шляхів вибирається той, який найбільш зручний для викладачів та прослідкувати їх навантаження. Оскільки цих варіантів скінченна кількість, то їх аналіз можна виконати вручну. Водночас, виходячи з інформації про кваліфікацію викладачів, їх ведення занять, рівень підготовки студентів та інших характеристик, можна обрати найкращий шлях реалізації виконання завдань у вибраній моделі системи. Наприклад, реалізацію першого завдання можна виконати послідовністю S_2, S_4, S_4, S_6 або послідовністю S_4, S_4, S_2, S_6 , а також послідовністю S_4, S_2, S_4, S_6 , яка, очевидно, не є зручною для викладача S_4 .

Друге, можливість полегшити роботу методистів як кафедри, так і методистів деканату через внесення додаткової інформації в графічне зображення. Ця інформація вноситься на випадок зриву виконання якоїсь операції (наприклад, неможливість фізичної присутності виконавця на

місці праці, затримка транспортних засобів у корках, хвороба виконавця операції, відсутність студентів в аудиторії тощо).

Така модифікація графічного зображення виконання завдань набуває вигляду:

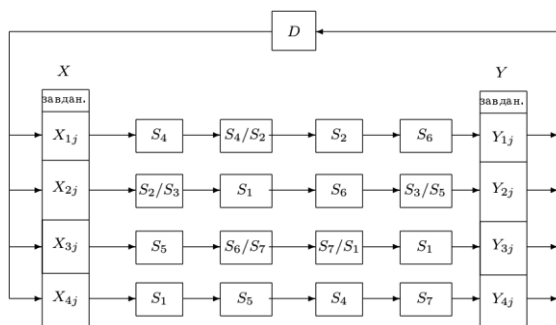


Рис. 3. Модифіковане графічне зображення розкладу виконання завдань j -го дня

У модифікованому таким чином графіку вказуються можливі заміни виконання операцій іншими виконавцями, які в стані повноцінно виконати операцію і можливо завдання в цілому.

$$A_{kj} = (A = \{D, X_{kj}, W_{kj}, Y_{kj}, Sh_{kj}, w_{kj}^1, w_{kj}^2, w_{kj}^3, w_{kj}^4\}, Z_{kj} = \{b, b_{kj}, s_{kj}, t_{kj}, a_{kj}^1, a_{kj}^2, a_{kj}^3, a_{kj}^4, b_{kj}^1, b_{kj}^2, b_{kj}^3, b_{kj}^4\}, D, f_{kj}, D),$$

де

- стан W_{kj} зображує ланцюжок станів виконання операцій $w_{kj}^1, w_{kj}^1, w_{kj}^1, w_{kj}^1$;
- стани X_{kj}, Y_{kj} зображують вхідний і вихідний сервери k -го завдання;
- Z_{kj} – вхідний алфавіт автомата;
- стан w_{kj}^r зображує стани виконання операції $w_{kj}^1, r = 1, 2, 3, 4$;
- стан Sh_{kj} зображує планування виконання аварійно закінченого завдання $w_{kj}^r, r = 1, 2, 3, 4$;
- символи b, b_{kj}, s_{kj} означають початок виконання і успішне виконання завдання відповідно;
- символи $a_{kj}^1, a_{kj}^2, a_{kj}^3, a_{kj}^4, b_{kj}^1, b_{kj}^2,$

b_{kj}^3, b_{kj}^4 означають аварійне закінчення r -ї операції і повторне виконання цієї операції відповідно.

Третє, наявність такого графічного зображення полегшує працівникам деканату розробку розкладу роботи як кафедри, так і факультету в цілому. Останнє потребує графічних зображень від усіх кафедр факультету. Водночас методисти дістають змогу розробити розклад з мінімізацією кількості викладачів протягом робочого дня у разі дефіциту аудиторного фонду та технічного обладнання.

Автоматно-мережева модель екосистеми

Оскільки виконання завдань в системі відбувається паралельно, то будемо моделювати процес виконання цих завдань мережею автоматів, яка є підходящою математичною моделлю для таких процесів [2]. Спочатку розглянемо автоматне зображення виконання одного завдання в системі.

Автомат, який моделює виконання k -го завдання, набуває вигляду:

Початковим і заключним станом автомата є стан D , а функція переходів f_{kj} зображена графом переходів на рис. 4.

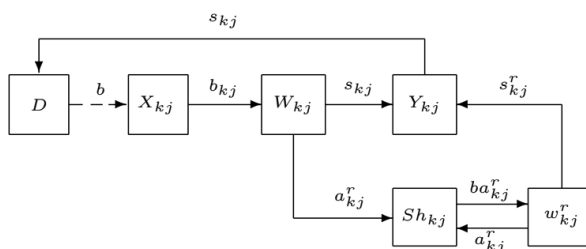


Рис. 4. Автомат A_{kj} виконання k -го завдання j -го дня

Тепер можна представити виконання завдань j -го дня роботи кафедри такою мережею із автоматів:

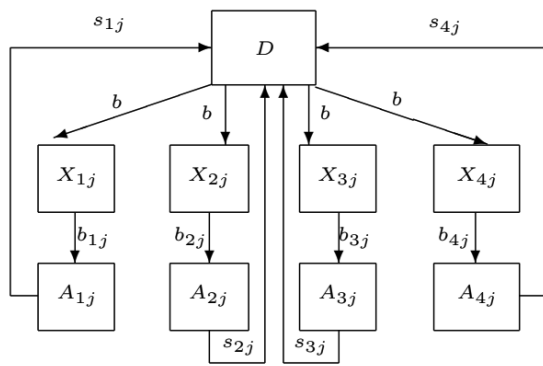


Рис. 5. Мережа з автоматів, яка моделює роботу j -го робочого дня кафедри

Висновки

Як впливає зі сказаного, всі процеси, задіяні в моделі, можуть бути автоматизовані. А це дозволяє як в ручному, так і в дистанційному режимі виконувати управління таким об'єктом, як "Кафедра". Зрозуміло також, що цю модель можна застосувати до моделювання роботи факультету і ЗВО в цілому.

Література

1. Kryvyi S., Opanasenko V., Grinenko E. Logical approach to the researcher of properties of software engineering ecosystem. *11th Int. IEEE Conf, on Dependable Systems, Services and Technologies. (IEEE DESSERT 2020)*. – Kyiv, 2020, pp. 456-464.
2. Кривий С.Л. Скінченні автомати: теорія, алгоритми, складність. - Чернівці-Київ: Букрек, 2020. - 427 с.
3. Baader F., Horrocks I., Carsten L., Sattler U. An Introduction to Description Logics. - Cambridge University Press, 2017. - 255 p.
4. Balmelli L., Brown D., Cantor M., Mott M. Model-driven systems development. *IBM Systems Journal*, 2006. V. 45. Pp. 569-585.
5. Reisig W. Petri nets. An introduction. *Springer Verlag: Berlin Heidelberg*, 1985. 161 p.
6. Arnold A. Finite Transition Systems: Semantics of Communicating Systems. - Paris: Prentice Hall, 1994. - 177 p.

References

1. Kryvyi S., Opanasenko V., Grinenko E. Logical approach to the researcher of properties of software engineering ecosystem. *11th Int. IEEE Conf, on Dependable Systems, Services and*

Technologies. (IEEE DESSERT 2020). – Kyiv, 2020, pp. 456-464.

2. Kryvyi S. *Skinchenni avtomaty: teoriya, algoritmi, skladnist.* – Chernivtsi-Kyiv: Bukrek, 2020. - 427 p.
3. Baader F., Horrocks I., Carsten L., Sattler U. *An Introduction to Description Logics.* - Cambridge University Press, 2017. - 255 p.
4. Balmelli L., Brown D., Cantor M., Mott M. *Model-driven systems development. IBM Systems Journal*, 2006. V. 45. Pp. 569-585.
5. Reisig W. *Petri nets. An introduction.* *Springer Verlag: Berlin Heidelberg*, 1985. 161 p.
6. Arnold A. *Finite Transition Systems: Semantics of Communicating Systems.* - Paris: Prentice Hall, 1994. - 177 p.

Одержано: 05.04.2024

Внутрішня рецензія отримана: 17.04.2024

Зовнішня рецензія отримана: 22.04.2024

Про авторів:

¹Кривий Сергій Лук'янович,
доктор фізико-математичних наук,
професор
<https://orcid.org/0000-0003-4231-0691>

²Грінченко Олена Олександрівна,
кандидат технічних наук,
<https://orcid.org/0000-0001-9673-6626>

Місце роботи авторів:

¹Київський національний університет імені Тараса Шевченка,
тел. +38-097-334-60-56
E-mail: sl.krivoi@gmail.com
<https://knu.ua/>

²Київський національний університет імені Тараса Шевченка,
Тел. +38-097-912-59-20
E-mail: olena.grinenko@knu.ua,
<https://knu.ua>