

## Правова інформатика

УДК 004.451.23:004.77

НЕСТЕРЕНКО О.В., НЕТЕСІН І.Є., Міжнародний науковий центр  
технології програмування “Технософт”

### ПРО ІНТЕРОПЕРАБЕЛЬНІСТЬ КОМПОНЕНТІВ АВТОМАТИЗОВАНИХ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ

*Анотація.* До розв’язання проблем регулювання взаємодії інформаційних процесів, що протікають в автоматизованих аналітичних системах органів державної влади, а також їх синхронізації стосовно доступу до ресурсів.

*Ключові слова:* інформаційний процес, автоматизована система, взаємодія, регулювання, синхронізація.

*Аннотация.* К решению проблем регуляции взаимодействия информационных процессов, которые протекают в автоматизированных аналитических системах органов государственной власти, а также их синхронизации относительно доступа к ресурсам.

*Ключевые слова:* информационный процесс, автоматизированная система, взаимодействие, регуляция, синхронизация.

*Summary.* To resolving problems of adjusting of informative interprocess, which flow in the automated analytical systems of public authorities, and also their synchronization in relation to access to the resources, communication.

*Keywords:* informative process, automated system, cooperation, adjusting, synchronization.

**Постановка проблеми.** Однією з характерних особливостей інформатизації суспільства є забезпечення вичерпною інформаційною підтримкою аналітичної діяльності та прийняття рішень в управлінських структурах і, як наслідок, активізація інформаційного обміну. Тенденцією розвитку інформаційно-комунікаційних технологій (далі – ІКТ), що постійно зростає, є формування гетерогенного середовища, в якому відбувається обмін інформацією. Ці особливості напряду пов’язані з вирішенням завдань формування й розвитку складових інформаційного суспільства, насамперед інформаційної системи електронного уряду. У зв’язку із цим в наш час усе більша увага приділяється питанням забезпечення інтероперабельності для систем різного масштабу і різного призначення. Згідно зі стандартом ISO/IEC 24765 під інтероперабельністю розуміють здатність двох або більше систем або компонентів системи до обміну та використання інформації. Властивість інтероперабельності, поряд із властивостями переносимості та сумісності, становить одну з найважливіших ознак відкритих систем.

Виходячи з ієрархії інформаційних систем різного масштабу, коли у кожній компоненті виділяють два типи інтерфейсів – “горизонтальні” (для зв’язку усередині компоненти) і “вертикальні” (для зв’язку між компонентами різного рангу), для кожної компоненти із цього ланцюжка існує власна проблема інтероперабельності, для вирішення якої розробляються рекомендації на підставі архітектурного підходу і стандартів.

Забезпечення інтероперабельності є фундаментальною основою процесу створення та розвитку автоматизованих інформаційно-аналітичних систем (далі – АІАС) різного призначення, особливо для АІАС органів державної влади, які працюють в умовах

оперативного інформаційного обміну між системами різних органів влади. Але слід констатувати, що систематичних робіт із забезпечення інтеперабельності в країні зараз не проводиться. Це ускладнює забезпечення своєчасності і обґрунтованості рішень, що, враховуючи їх державне значення, загрожує безпеці країни.

Специфіка взаємодії в АІАС, яка має важливе значення для їх проектування, пов'язана з процесами передачі та обробки документів, а також з тим, що під взаємодією варто розуміти процес впливу різних об'єктів та процесів системи один на одного, що приводить до зміни стану їх інформаційних ресурсів та особливостей опрацювання. Тому актуальними є задачі формалізації процесів інформаційної взаємодії, визначення основних засад регламенту такої взаємодії, особливостей забезпечення управління інформаційними ресурсами, а також забезпечення інформаційної безпеки.

У якості бази вирішення цих проблем у [1] запропоновано основи нової теорії ситуаційного регулювання технологічних процесів обробки інформаційних потоків в органах влади, що має забезпечити інтеперабельність документообігу та опрацювання інформації в умовах послідовно-паралельного підключення для підготовки проектів рішень різних експертів, підрозділів та органів влади.

Необхідність керування автоматизованою системою (далі – АС) завдяки зміні структури, використання адаптивної структури, зміні напряму передачі інформаційних потоків та ін. в рамках ситуаційного регулювання ставить цілий ряд складних, специфічних задач, більшість з яких дотепер ще не вирішено, а частина з них навіть не сформульована належним чином. Тому методи аналізу, синтезу й оптимізації таких систем здобувають виняткового значення, а разом із цим й відповідні методології, що дозволяють розв'язувати окремі задачі досліджень за вказаними напрямками.

**Метою статті** є розробка опису процесів у локальному та розподіленому середовищах за умов здатності до взаємодії (інтеперабельності) компонентів автоматизованих інформаційно-аналітичних систем.

#### **Виклад основних положень.**

*Підхід до опису взаємодіючих процесів.* При вивченні проблеми взаємодії, яке відбувалось у процесі створення АС разом зі зміною парадигми *обчислень* на парадигму *взаємодії*, пов'язаною з переходом від великих машин до сукупності робочих станцій, серверів і комп'ютерних мереж, багато авторів давали їй різні тлумачення. Так, згідно з [2, 3] обчислення й взаємодія – це дві ортогональні концепції, тобто взаємодія розглядається як деяка дія, а не обчислення.

Порівняно з традиційними алгоритмами обчислень поняття взаємодії процесів і об'єктів у розподіленому середовищі, яким, власне, є й АІАС, стали вважати більш сильною концепцією, тому що за його допомогою можна виразити в сучасних системах процеси дій (наприклад, рух або перетворення електронних документів). Локальні процеси, що породжені локальним програмним середовищем та відбуваються у вузлах АІАС, як правило, впливають не на загальний стан і поведінку системи, а тільки на поведінку програмного середовища, яке виконує визначену послідовність дій для обчислень деякого значення функції, що реалізується. У той же час розподілені взаємодіючі процеси, що виконуються під впливом обміну повідомленнями між вузлами (локальними й віддалено розташованими), можуть впливати на загальний стан і поведінку системи (наприклад, переповнення буфера мережного протоколу, перевантаження серверів, відмова каналів передачі даних та ін.).

Розглянемо взаємодію інформаційних процесів на прикладі АІАС органу державної влади (далі – ОДВ). Передусім визначимось, що такі системи є “людино-машинними”, тобто співробітники ОДВ, які беруть участь в опрацюванні документів (будемо називати

їх експертами), є не просто користувачами, але являють невід’ємну складову процесів, що протікають в АІАС.

У загальному випадку при надходженні в ОДВ документа ініціюється процес його виконання (або обробки), одним або декількома експертами з використанням автоматизованих робочих місць та доступами до поділюваних ресурсів. Враховуючи достатньо тривалий час протікання цих процесів (адже на виконання документа може відводитись до 30 календарних днів), за цей період у сучасних умовах постійного зростання динаміки суспільно-політичних процесів чимало може змінитися, і намічений експертом план виконання документу може зазнати суттєвих змін. На процес, що розглядається, також може вплинути й результат розгляду іншого документа, що паралельно ініційовано цим же експертом або деяким іншим та який отримано до завершення вказаного процесу.

Таким чином, процеси опрацювання документів можуть протікати послідовно й паралельно, бути залежними або незалежними один від одного, рівноправними або привілейованими, відкритими або захищеними, виконуватися синхронно або асинхронно й тощо, тобто протікання та взаємодії процесів є багатоаспектними.

Більш того, незважаючи на те, що усі підсистеми АІАС є детермінованими, тобто такими, для яких кожному значенню вхідних дій відповідають цілком визначені значення вихідних даних, уся АІАС, враховуючи можливість випадкових зв’язків між підсистемами, вже відноситься до класу недетермінованих систем. Дійсно, маршрут опрацювання документа може проходити різними шляхами залежно від того, до яких експертів він потрапить (наприклад, після очікування в черзі – до експерта, який є фахівцем потрібного напрямку і вивільнився першим), а також які інформаційні ресурси будуть використані цими експертами [4]. Також часто-густо документ одночасно паралельно надається різним експертам, після чого відбираються кращі пропозиції.

Тому моделювання таких взаємодіючих процесів пов’язане з багатьма труднощами, адже вони визначаються не тільки названими особливостями виконання процесів, а й їх локальністю або розподіленістю.

Використовуючи теоретико-множинне визначення системи, узагальнену модель системи  $C$  можна описати виразом:  $C = \langle V, P, A, T \rangle$ ,

де:  $V$  – вузли системи, під якими розуміються АРМ експертів та засоби (сервери), що містять ресурси;  $P$  – цілісний процес функціонування системи як набір взаємодіючих процесів  $P = \{P_k\}$ ,  $k = 1, 2, 3, \dots$  – порядковий номер процесу (наприклад, у межах календарного року);  $A$  – множина можливих дій, що впливають на процеси  $P_k$ ;  $T$  – часові упорядкування, залежності, обмеження при активізації дій та виконанні процесів. Залежно від технології взаємодії процесів це може бути календарний час, інтервальний час, вузловий час та ін. [5].

Вузли можна описати як  $V = \{S, \Delta\} = \{V_l\}, l = 1, \dots, N + R$ , де:  $S = \{S_j\}, j = 1, \dots, N$  – множина експертів,  $\Delta = \{\Delta_r\}, r = 1, \dots, R$  – множина ресурсів. Тобто під системою будемо розуміти сукупність вузлів і засобів мережної взаємодії, що зв’язують інформаційно будь-яку пару вузлів системи між собою. Вузол системи  $V$  – це самостійний її компонент, що містить ресурсні об’єкти і компоненти – технічні (АРМ експерта, сервер і т.п.), програмні (ОС, мережні процедури, прикладні програмні компоненти й ін.) та інформаційні (БД, локальні засоби обробки інформації та управління, файли та ін.).

Для системи  $C$  процес  $P_k$  – це повний цикл опрацювання документа від його отримання до відправлення результату адресату. Процеси нумеруються у порядку породження. У даному викладенні під взаємодіючими процесами будемо розуміти процеси опрацювання документів  $d_i$ , що ініційовані експертами  $S_j$ , тобто  $P_k \equiv P_{ij}$ .

Дії  $A = \{A_k^m\}, k, m \in z$ , тобто  $A_k^m$  – це дія, що породила новий стан  $m$  процесу  $P_k$ . При цьому  $A_k^m(V_i)$  або  $A_k^m(P_k)$  – відповідні дії, активовані вузлом  $V_i$  або процесом  $P_{k+1}$ , що породили новий стан  $m$  процесу  $P_k$ . Діями, наприклад, є передача документа від одного експерта до іншого, відправлення запиту до ресурсу, відправлення повідомлення, поставлення документа в чергу до експерта (ресурсу), переривання процесом з більшим пріоритетом процесу з меншим пріоритетом та ін.

Таким чином, процеси  $P_k$  ініціюються у вузлах, протікання та взаємодія яких характеризується Рис. 1. При цьому показано:

- обробка інформації (документів)  $d_i$  експертами  $S_j$  (Рис. 1 а);
  - пошук та обробка інформаційних ресурсів  $Q_i$  (структурованих даних з БД, аналітичних довідок та ін.) (Рис. 1 б);
  - активізація дій  $A_{k,k+1}$  процесу  $P_k$  на процес  $P_{k+1}$  (Рис. 1 в);
  - керування взаємодією  $C_{k,k+1}$  процесів (рис. 1 в),
  - часове обмеження на проходження процесу  $P_k$ ,
- де:  $k, k+1$  – номери взаємодіючих між собою процесів.

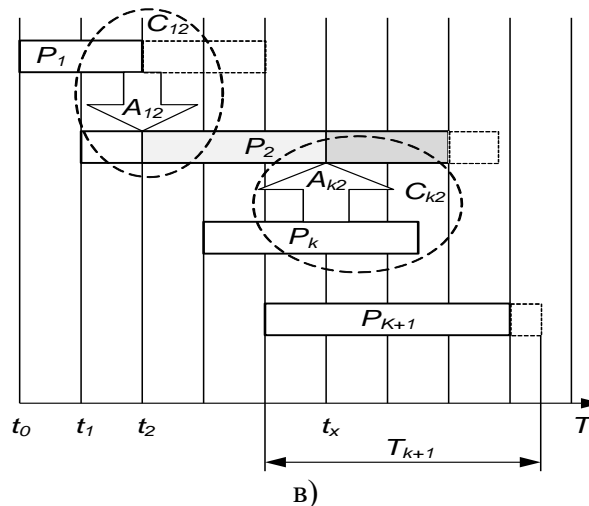
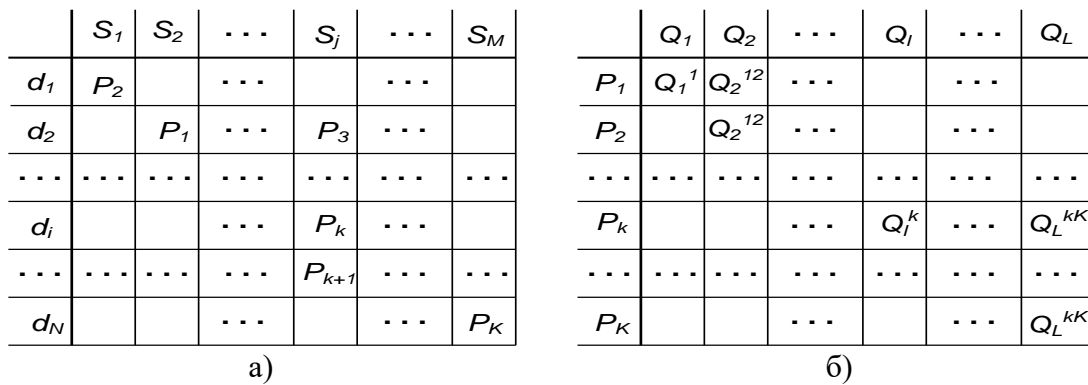


Рис. 1. Протікання та взаємодія процесів (а – породження процесів, б – використання процесами ресурсів, в – дії процесів один на одного та керування взаємодією)

Як видно з наведеного рисунку, процеси можуть бути пов'язаними з опрацюванням одним експертом різних документів, одного документа різними експертами, одночасним використанням одного ресурсу двома (і більше) процесами ( $Q_L^{kk}$  на рис. 1 в), діями одного процесу на інший як по закінченню першого, так і в процесі його протікання. Також слід

зазначити, що дії  $A_{k,k+1}$  переводять процес  $P_{k+1}$  з одного стану  $z_u^{P_{k+1}}$  в інший стан  $z_{u+1}^{P_{k+1}}$  (Рис. 1).

Виходячи з наведених визначень модель взаємодії процесів  $k$  та  $k+1$  ( $\Theta_{k,k+1}$ ) має вигляд:

$$\Theta_{k,k+1} = \{P, B, Q, T\},$$

де:  $P = \{P_k, P_{k+1}\}$  – параметри процесів  $P_k$  та  $P_{k+1}$ ;

$B = \{A\{A_{k,k+1}\}, C\{C_{k,k+1}\}\}$  – процес взаємодії процесів  $P_k$  та  $P_{k+1}$ ;

$Q = \{Q_j^k, Q_j^{k+1}\}$  – інформаційні ресурси, потрібні для підтримки процесів  $P_k$  та  $P_{k+1}$ ;

$T = \{T_k, T_{k+1}\}$  – часові характеристики процесів  $P_k$  та  $P_{k+1}$ .

Дії, що впливають на процеси та їх взаємодію за моделлю  $\Theta_{k,k+1}$  та підтримуються вузлами у мережному середовищі, як правило, ініціюються повідомленнями. Кожний вузол під впливом розподілених дій, що виходять від процесів інших вузлів, ініціює перехід своїх процесів у новий стан, результати якого відображаються в розподіленому розділеному стані (*shared state – ss*) ресурсів (пам’яті) [6]. Обмін повідомленнями визначається виконанням послідовності операцій (*op*) деяких локальних програм у вузлах на стані розділеної пам’яті (Рис. 2).

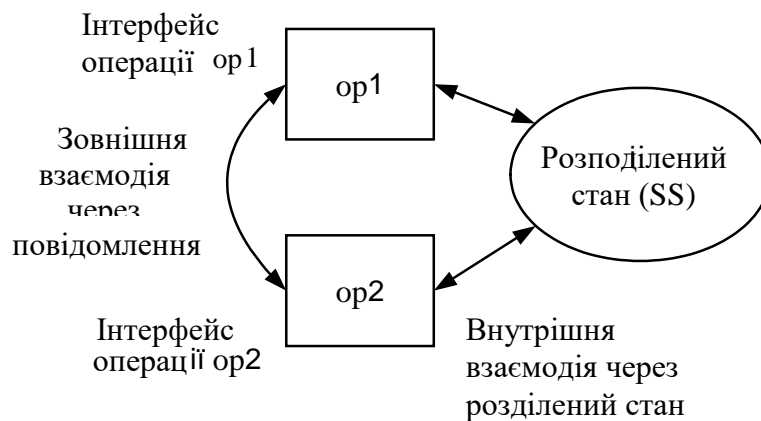


Рис. 2. Інтерфейс взаємодії операцій

*Формалізація взаємодії процесів.* У цей час склалась низка підходів до формального опису взаємодій об’єктів та процесів [2, 3, 7, 8]. Для формалізації запропонованої моделі  $\Theta_{k,k+1}$  (без обліку фактора часу) найбільш придатною, на наш погляд, є теорія мов дій (*Action Languages – AL*) [9], що є орієнтованою на формальний опис обчислювальних систем з паралельними елементами, а також на опис дій у розподілених середовищах. При цьому з метою абстрагування поняття ресурсів і пам’яті в *AL* ототожнюються.

Основа цієї теорії становить дія (*action*), за допомогою якої трансформується локальний або глобальний стан деякого об’єкту (процесу) в обчислювальному середовищі. Дії в мові *AL* можуть бути примітивні й складні, як комбінація примітивних. Значення дій визначаються деякою алгеброю дій (*Action Algebra – AA*).

Головною операцією алгебри є бінарна асоціативна операція, що має назву комбінації дій  $a$  і  $b$  (позначається як  $a \times b$ ). Враховуючи, що алгебра *AA* є комутативним моноїдом, у який можна включати й інші операції, у [7] розглядаються операції недетермінованого вибору з  $a$  і  $b$  (позначається як  $a + b$ ) і послідовна композиція дій (позначається  $ab$ ).

Поняття недетермінованого вибору є для процесів, що розглядаються, досить важливим, адже певна невизначеність, яка має місце при опрацюванні документів, може випадково зменшуватися або навіть збільшуватися під впливом взаємодіючих процесів у зв’язку із вищезазначеними особливостями виконання документів (процесів) в ОДВ.

Таким чином, кожний процес має власну поведінку, яка представляється маршрутом слідування документів  $M(d_k)$  (Рис. 3), тобто поведінка процесу  $B(P_k) \equiv M(d_k)$ . Для цього рисунку використані дані (Рис. 1 а). На рисунку момент, коли на процес  $P_2$  одночасно впливають два процеси (дії  $A_{12}$  та  $A_{k2}$ ) відповідає ситуації недетермінованого вибору.

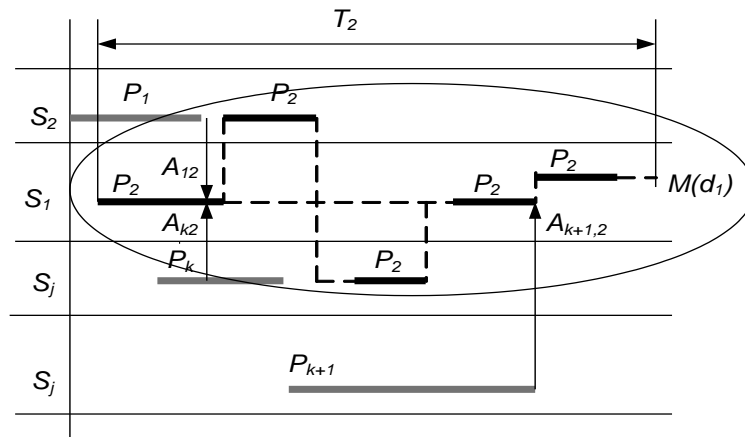


Рис. 3. Поведінка процесу  $P_2$

Поведінка процесу може бути елементом алгебри поведінки над  $A$ . Ця алгебра має дві операції. Перша (позначається як  $u + v$ ) є внутрішньою бінарною асоціативною і комутативною операцією, що відповідає недетермінованому вибору. Друга операція – префіксна (позначається як  $au$ ), де:  $a$  – дія;  $u$  – поведінка.

Тоді з самої природи алгебри дій слідує, що кожна поведінка процесу може бути представленою у формі:  $u = \sum a_i u_i + e, i \in I$ ,

де:  $a_i$  – ненульові детерміновані дії;  $u_i$  – поведінка;  $I$  – скінчена або лічильна множина індексів;  $e$  – термінальна константа (мінімальний елемент поведінки, пуста поведінка, неможлива поведінка).

Модель взаємодії об’єктів системи та аналіз теорії мов дій у застосуванні до розподілених комп’ютерних мереж наведено у [10], де показано, що  $AL$  дозволяє визначити інтерактивну семантику програм з локальною пам’яттю та їх поведінку в умовах змінного середовища та розширити семантику  $AL$  програм з локальною пам’яттю розподіленою інтерактивною семантикою віддалених програм з загальною глобальною пам’яттю (ресурсом).

Основою цієї семантики є механізм семантики дій  $AL$  програм з локальною пам’яттю, трансформований у розподілену мову дій  $DAL$  (*Distributed Action Language*), що використовується для формування стану загальної пам’яті при виконанні віддалено розподілених компонент з двома основними операціями – недетермінованого вибору, за допомогою якої вибирається поведінка програми, та префіксною операцією, яка визначає результат деякої дії на поведінку програми.

Застосовуючи ідеологію  $AL$ -мови для нашого середовища, тобто системи  $S$ , і враховуючи, що поведінка  $B(P_k)$  процесу  $P_k$  в деякий момент не залежить від наступних дій, тобто процеси володіють префіксною властивістю, отримуємо:

$$B(P_k) = \sum_{m \in Z} A_k^m * P_k^m + P_0,$$

де:  $\sum$  – операція недетермінованого вибору поведінки процесу; \* – префіксна операція дії на процес;  $P_0 = B(\emptyset)$  – пустий процес, тобто з відсутністю поведінки.

Розглянута теорія мов дій, у тому числі мови DAL, дозволяє провести аналогію між основними поняттями DAL і моделі (2), а саме:

Терміни мови DAL ←	→ Терміни моделі (2)
Програма	Процес
Пам'ять	Ресурс (об'єкт)
Дія	Метод або функція, що ініційовані процесом
Поведінка	Маршрут слідування документа
Середовище	Конкретне середовище системи

Проведений аналіз показує, що існує загальність у визначенні окремих понять і складається впевненість у можливості формалізації моделі (2) засобами мови AL.

*Систематизація знань про взаємодіючі процеси.*

Під систематизацією знань про взаємодію в АІАС розуміється виділення взаємодіючих процесів і об'єктів, їхніх базових властивостей і понять; визначення моделей подання знань про процеси і об'єкти середовищ і розподілених додатків, що занурюються в них; створення репозиторія знань як понятійної бази взаємодіючих процесів середовища.

При поданні знань процеси поділяються на дві категорії. До першої з них відносяться класи процесів, що відповідають системним і сервісним об'єктам забезпечення взаємодії. До процесів другої категорії відносяться класи процесів, що підтримуються прикладними об'єктами (локальними або віддаленими додатками).

Відомості про процеси першої категорії, як правило, представляються у вигляді фреймової моделі й графа відношень. При завданні процесів другої категорії вказується сукупність знань про кожний процес зазвичай у вигляді набору фасет, що відображають реалізовану функцію, вид специфікації, тип взаємодії з іншими процесами, середовище функціонування й ін. Ці дані уміщаються у репозиторій знань і використовуються при видобуванні знань про ці процеси.

Використовуючи ідеї й положення алгебри мов дій, можна з нових позицій підійти до систематизації процесів моделі  $\Theta_{k,k+1}$ , а саме – за критерієм їхньої реакції на певні дії.

Однією з концептуальних засад АІАС органів влади є забезпечення “виконавчої обов'язковості” (далі – ВО) документів [11]. При розгляді цих питань інтерпретується широко відома аксіома: всі питання забезпечення ВО описуються доступами суб'єктів до об'єктів. У цьому викладенні під суб'єктом будемо розуміти процес, що породжений деяким експертом (суб'єктом  $S_j$  виконання документа), а під об'єктом – інші процеси або ресурси деякого вузла. Множину можливих доступів будемо позначати  $R$ . Стосовно доступу до процесу визначимо можливими доступ на пересилання повідомлення ( $s$ ), призупинення процесу ( $p_0$ ) або відміни призупинення ( $p_1$ ), визначення стану процесу ( $z$ ), а до об'єкта – доступ на запис ( $w$ ), читання ( $r$ ), пересилання ( $m$ ), видалення ( $d$ ), копіювання ( $c$ ), тобто  $R = \{s, p, z, w, r, m, d, c\}$ .

Дія над процесом виконується через посилку запиту на обчислення методу процесу й контакту цього процесу з відповідними службами обробки запиту в середовищі системи. Основу контакту для забезпечення взаємодії процесів становить інтерфейс, що

складається з набору операцій, які сприяють утворенню стану його пам'яті в динаміці виконання програмного компонента. На зовнішньому рівні операції інтерфейсу задають набір дій неалгоритмічного типу, що викликають ініціацію деяких видів дій з обробки процесу в класі розподілених засобів. Даний підхід до класифікації можна розвивати, вводячи різні умови на порядок дій, їхню контекстну залежність, паралельність і т.д.

#### *Синхронізація інформаційних процесів.*

Однією з проблем інтегрованих систем є визначення методів синхронізації процесів, що конкурують за доступ до *загальних поділюваних ресурсів*.

У обставинах, коли суттєво зростає кількість документів для опрацювання з одночасними вимогами до скорочення термінів прийняття рішень, може виникати інформаційне “наднавантаження” [4]. У таких випадках потрібно забезпечувати найбільш повне і своєчасне отримання державних експертів необхідною інформації, а також ефективне використання усіх ресурсів АІАС (серверів, АРМ, інформаційних ресурсів тощо). Для подолання виникаючих при цьому конфліктних і позаштатних ситуацій, здатних призвести до недоступності потрібної інформації, зниженню працездатності системи, несвоєчасності виконання завдань, доступ до загального ресурсу, поділюваному між кількома користувачами (процесами), вимагає *синхронізації*, що полягає в координації та часовому узгодженні процесів.

*Поділюваним* називається загальний ресурс автоматизованої системи (база даних – БД), таблиця БД, окремий запис БД, файл з електронним документом тощо), з яким відразу можуть працювати кілька процесів. Поділюваний ресурс, з яким у цей момент часу може працювати тільки один процес, зазвичай називають *критичним*. Період, протягом якого процес працює із критичним ресурсом, є його *критичним інтервалом*. Будемо говорити, що процес  $P$  перебуває у своєму *критичному інтервалі* щодо деякого поділюваного ресурсу  $q$ ,  $q \in Q$ ,  $Q$  – сукупність поділюваних ресурсів АІАС, якщо в цей момент він захопив цей ресурс і працює з ним.

За сучасного розвитку продуктивності обчислювальних засобів та їх обсягів пам'яті конфліктні ситуації при доступі до ресурсів автоматизованої системи виникають доволі рідко, але для ОДВ, де вага державного рішення буває досить суттєвою, навіть окремі випадки затримання обробки можуть бути критичними.

Зазвичай синхронізація процесів при зверненні до поділюваного ресурсу забезпечується механізмами *блокування*, постановки запитів у *чергу* й *взаємного виключення доступу* до поділюваного ресурсу. Цю суть блокування пропонується поширити й на синхронізацію взаємодії процесів, що розглядаються, тобто на час роботи у деякому процесі будь-якого експерта як “поділюваного ресурсу” доступ до нього з боку інших процесів (експертів) тимчасово забороняється або обмежується.

Метод блокування [12, 13] спочатку був розроблений для поділу загальної оперативної пам'яті між програмними процесами, що одночасно протікають, однак він залишається актуальним і для взаємодіючих процесів, які спільно використовують й інші загальні ресурси. В основі цього методу лежить поняття семафора (*semaphore*) як замка доступу до зайнятого ресурсу. За його допомогою поділюваний ресурс, що використовується протягом деякого часу одним процесом, оголошується на цей проміжок часу закритим для інших процесів.

В АІАС, яка функціонує у документальному середовищі, розглянутий механізм не вирішує проблеми синхронізації. Специфіка такого середовища полягає у тому, що документи або відправлені процесами повідомлення можуть прибувати до одержувача (експерта) з часовими затримками, якими не можна зневажати, і часто не в тому порядку, у якому вони відправлялися, що впливає на черговість опрацювання. Тому



методи синхронізації в АІАС мають враховувати часові характеристики повідомлень (запитів), що відправляються. Відомий метод взаємного виключення (*mutual exclusion*), зокрема для мережного середовища [14, 15], забезпечує в кожний момент часу право володіння поділюваним ресурсом тільки одному процесу. Інші процеси при цьому не конкурують за цей ресурс доти, поки процес, що захопив ресурс, не звільнить його. Алгоритми реалізації методу взаємного виключення використовують як параметри повідомлень порядковий номер повідомлення або штамп часу, а також номер вузла.

Зазначені чинники методу взаємного виключення можливо поширити й на випадок взаємодії процесів, що розглядається. Синхронізація процесів має будуватись на постійній активній взаємодії між собою всіх процесів, що конкурують за доступ до загального ресурсу (експерта), і тому має спиратися на наступні два основні принципи:

- 1) процес  $P_k$  від деякого експерта  $S_j$  входить у свій *kinp*  $P_k(S_j)$  тільки після того, як він повідомив про це всім іншим процесам й одержав від них відповіді-дозволи;
- 2) якщо процес  $P_k$  повідомив всі інші процеси про свій намір увійти в *kinp*  $P_k(S_j)$ , жоден з процесів від інших експертів не може увійти у свій *kinp* щодо цього ж експерта  $S_j$  двічі, поки  $P_k$  не реалізує свою вимогу.

Крім головної мети – синхронізації роботи процесів, ставиться й розв’язується не менш важлива задача попередження виникнення двох небажаних станів у системі: *тупикового* стану, за якого жоден процес  $P$  не перебуває у своєму *kinp*  $P(S_j)$  і жоден інший процес, що бажає увійти у свій *kinp* щодо цього ж  $S_j$ , не може цього зробити, і стану *голодування*, за якого деякий процес  $P$  невизначено довго чекає, щоб увійти у свій *kinp*  $P(S_j)$ , у той час як інші процеси постійно входять і виходять зі своїх *kinp* щодо цього ж  $S_j$ .

Для розв’язання задачі синхронізації процесів, що запитують поділювані ресурси в АІАС, пропонується підхід, заснований на синхронізуючому автоматі (СА) [5]. СА є програмою, розташованою в кожному вузлі АІАС, що є АРМом експерта та циклічно працює за певним алгоритмом. Функціонування СА відповідає циклу дискретного автомата [16]: читання вхідного сигналу, вироблення вихідного сигналу (результату), зміна стану.

Для реалізації задачі впорядкування запитів і синхронізації відповідних їм процесів задамо частковий СА для вузла  $V_j$  у такий спосіб:

$$CA(V_j) = \langle A_j, B_j, S_j, \varphi, \psi, s_j^0, \pi \rangle, \text{ (див. Рис. 4).}$$

- де:  $A_j$  – вхідний алфавіт СА;
- $B_j$  – вихідний алфавіт СА;
- $S_j$  – множина станів СА;
- $\varphi: A_j \times S_j \rightarrow B_j$  – функція виходів;
- $\psi: A_j \times S_j \rightarrow S_j$  – функція переходів;

$s_j^0 \in S_j$  – початковий стан СА;

$\pi \in S_j$  – спеціальний стан.

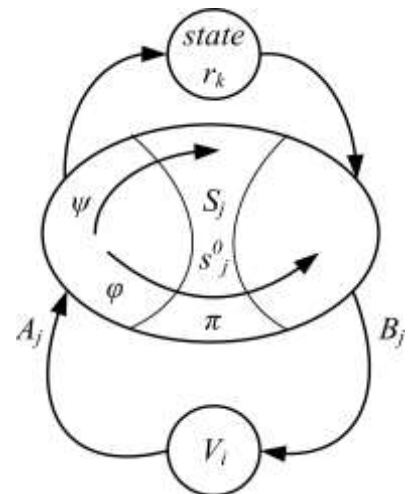


Рис. 4. Графове представлення СА вузла  $V_j$

Фактично  $A_j$  є множиною запитів, що надходять до вузла  $V_j$  з вузлів  $V_i$ , а  $B_j$  – множиною повідомлень-відповідей, що дозволяють використання ресурсу.

Проведені у [5] дослідження застосування концепції СА для розв’язання задачі синхронізації процесів при доступі до розподілених поділюваних ресурсів, зроблені висновки стосовно врахування протікання часу та часових міток (застосування лампорового часу, визначення інтервального часу для вузлів, посилення пустих

повідомлень (пухирців) та ін.) свідчать про перспективність застосування вказаного підходу для синхронізації процесів, що взаємодіють в АІАС.

### **Висновки.**

В сучасних умовах, коли здатність до взаємодії (інтероперабельність) компонентів АІАС, особливо в органах влади, та систем між собою є визначальним фактором формування інформаційного суспільства, актуальним є розгортання систематичних досліджень щодо вирішення проблем інтероперабельності.

Розглянутий підхід до формального опису взаємодії процесів у локальному та розподіленому середовищах, заснований на теорії мов дій, запропоновані модель взаємодії об'єктів АІАС та підхід до систематизації знань про процеси з використанням основних понять даної теорії, а також аналіз використання синхронізуючого автомата, що базується на принципах дискретних автоматів, для регулювання виконання процесів у розподіленому середовищі АІАС можна вважати визначенням деяких напрямів проведення вищевказаних досліджень.

### **Використана література**

1. Нестеренко О.В. Методологія ситуаційного регулювання в автоматизованих системах для забезпечення необхідного рівня інформаційної безпеки державної влади / О.В. Нестеренко / Реєстрація, зберігання і обробка даних. – 2008. – Т. 10. – № 4. – С. 25-36.
2. Wegner P. Why interaction is more powerful than algorithms / P. Wegner // SACM. – 1997. – 40. – № 45. – P. 80 – 91.
3. Wegner P. Interaction as a Basis for Empirical Computer Science / P. Wegner / ACM Computing Surveys. – 1995. – № 1. – P. 45 – 48.
4. Нестеренко О.В. Моделі інформаційного навантаження при опрацюванні документів в автоматизованих інформаційно-аналітичних системах органів державної влади / О.В. Нестеренко, І.Є. Нетесін // Реєстрація, зберігання і обробка даних. – 2011. – Т. 13. – № 1. – С. 39-51.
5. Нетесин И.Е. Подход к синхронизации процессов в компьютерных сетях / И.Е. Нетесин / Проблемы программирования. – 2001. – № 1-2. – С. 76-83.
6. Нетесин И.Е. Определение основ взаимодействия объектов в компьютерных сетях : труды Международной конференции УкрПрог-2000. – К., 2000, 23 – 26 мая.
7. Hoare C.A.R. Communicating Sequential Processes / C.A.R. Hoare. – Prentice Hall, UK, 1985.
8. Milner R. Elements of interaction. Turing Award Lect / R. Milner / Commun. ACM. – 1993. – № 1. – P. 78 – 89.
9. Letichevsky A.A. A general theory of action languages / A.A. Letichevsky, D.R. Gilbert / Кибернетика и системный анализ. – 1998. – № 1. – С. 16 – 36.
10. Нетесин И.Е. Интеллектуализация и систематизация знаний о взаимодействующих объектах компьютерных сетей / И.Е. Нетесин / Искусственный интеллект. – 2000. – № 2. – С. 179-86.
11. Нестеренко О.В. Інформаційний підхід до забезпечення керування в автоматизованих інформаційно-аналітичних системах органів влади / О.В. Нестеренко / Реєстрація, зберігання і обробка даних. – 2008. – Т. 10. – № 3. – С. 46 – 55.
12. Дейкстра Э. Взаимодействие последовательных процессов / Э. Дейкстра / Языки программирования. – М. : Мир, 1972. – С. 9 – 86.
13. Хоар Ч. Взаимодействующие последовательные процессы / Ч. Хоар. – М. Мир, 1989. – 264 с.
14. Lamport L. Time, Clocks and the Ordering of Events in a Distributed System / L. Lamport / Comm. ACM. – 1978. – № 7. – P. 558 – 565.
15. Ricart G. An Optional Algorithm for Mutual Exclusion in Computer Networks / G. Ricart, A.K. Agrawala // Comm. ACM. – 1981. – № 1. – P. 9 – 17.
16. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М. : Физматгиз, 1962. – 476 с.