УДК 004.75.03

В.О. Гожий

АЛГЕБРА ДЛЯ МОДЕЛЮВАННЯ ВЗАЄМОДІЇ WEB-СЕРВІСІВ

Анотація. Представлена алгебра для моделювання взаємодії web-сервісів. В якості основних базових конструкцій для алгебри використовуються: послідовність, альтернативний вибір та цикл. Крім того, визначено комбіновані конструкції, якими є паралельність, довільна послідовність, дискримінатор та динамічний вибір. Конструкції були обрані для забезпечення загального та розширеного набору операцій для комбінації web-сервісів. Після визначення кожної конструкції дається формальна семантика оператора в термінах мереж Петрі. Наведено приклад взаємодії web-сервісів.

Ключові слова:Алгебра, web-сервіс, взаємодія web-сервісів, мережі Петрі.

Вступ

Використання методів візуального представлення та моделювання, таких як мережі Петрі на етапі розробки складних технічних систем ефективні за декількаами причинами. По-перше візуальні представлення забезпечують високорівневу, але точну мову опису та представлення складної системи, яка дозволяє формально описувати та моделювати процес на різних рівнях абстракції. По-друге мережі Петрі дозволяють моделювати складні системи в динаміці. Одним із прикладів складних систем, які працюють динамічно, є веб-сервіси.

Веб-сервіс представляє собою компонент, написаний на будь-якій мові, який може бути розгорнутим на будь-якій платформі, яка має стандартний інтерфейс на основі XML. Поведінка регламентується структурою веб-сервисів, в основному це частково впорядкований набір операцій. Це робить їх зручними для відображення за допомогою мереж Петри. Операції моделюються переходами, а стани сервісу моделюються позиціями. Стрілки між позиціями та переходами використовуються для позначення причинно-наслідкових зв'язків [4,5]. Компоненти веб-сервісів можуть взаємодіяти з іншими додатками, які самі відповідають стандартам веб-сервісів. Як правило, один сер-

віс не задовольняє потреб користувачів, і сервіси стають все більш складними. Фактично сучасний веб-сервіс створюється шляхом поєднання різних веб-сервісів та їх компонентів для створення компонентного сервісу, який пропонує набір нових функціональних послуг. Цей процес називається композицією веб-сервісів [7].

При композиції веб-сервісів самим критичним є взаємодія веб-сервісів та їх компонентів між собою, що вимагає детального формального опису процесів функціонування та дослідження і моделювання їх поведінки. В роботі [7] описано алгебру для побудови компонентних сервісів, але в реалізації цієї алгебри недостатньо базових конструкцій, які не дозволяють реалізувати динамічну взаємодію сервісів. В цьому контексті пропонується стохастична алгебра для моделювання взаємодії веб-сервісів, та їх компонент на основі кольорових мереж Петрі, за допомогою якої провадиться формальний опис та моделювання складних композицій веб-сервісів. Базові та розширені конструкції, які підтримуються запропонованою алгеброю, синтаксично і семантично визначені та реалізовані при розробці нових веб-сервісів.

Постановка проблеми

Метою данної роботи є розробка сервісної алгебри для побудови моделей взаємодії веб-сервісів та їх компонентів на основі мереж Петрі, та їх модіфікацій.

Аналіз останніх досліджень

Використання мереж Петрі, як інструмента графічного і математичного моделювання складних систем та процесів останнім часом отримало широке розповсюдження [4,5,11]. Мережа Петрі (PN) це орієнтований двухдольний граф з двома типами вузлів: позиціями (представлені колами), та переходами (представлені прямокутниками). Дуги графа з'єднують позиції і переходи таким чином, що позиції можуть бути пов'язані тільки з переходами і навпаки. Позиції в мережі Петрі можуть містити дискретне число токенів. Розподіл токенів над місцями називається маркуванням. Коли К.А.Петрі вперше представив мережі Петрі, вони служили для опису паралельних систем з точки зору причинно-наслідкових зв'язків без обліку часу [11]. Впровадження часових концепцій в мережі Петрі було запропоновано декількома роками пізніше іншими дослідниками [6]. Для моделювання складних систем та процесів різної природи використовуються

різні модифікації мереж Петрі: стохастичні мережі Петрі (SPN), часові мережі Петрі (TPN), кольорові мережі Петрі (CPN) , нечіткі мережі Петрі (FuzzyPN) та інш. [2,6,8,9,10,13]. В роботах [7,12] мережі Петрі різних типів були використані для побудови різних веб-сервісів. Але задача моделювання взаємодії складних веб-сервісів та їх компонентів залишається актуальною. Для вирішення даної задачі розроблено алгебру сервісів із розширеним набором операцій, що дозволяє детально описати взаємодію між сервісами та їх компонентами при побудові і моделюванні складних веб-сервісів.

Формальний опис процесу функціонування сервісів

Взаємодія web-сервісів може бути ***статичною*** (коли компоненти сервисів взаємодіють по попередньо визначеному алгоритму), або ***динамічною*** (коли компоненти взаємодіють в процесі виконання операції, або процесу). В термінах мереж Петрі пропонується наступний опис.

Сервісна мережа являє собою позначену мережу, тобто це множина

$$SN = (P, T, W, i, o, l),$$

де

- P – кінцева множина позицій,
- T = IT ∪ TT - кінцева множина переходів, що представляють операції сервісів. IT – множина безпосередніх переходів, які схематично позначені чорними прямокутниками. TT – це множина часових переходів, що позначені порожніми прямокутниками.
- W ⊆ (P Ч T) ∪ (T Ч P) являє собою набір спрямованих дуг (напрям відносин),
- i - вхідна позиція, з $\cdot i = \{x \in P \cup T \mid (x, i) \in W\} = \varnothing$,
- o - вихідна позиція з $o \cdot = \{x \in P \cup T \mid (o, x) \in W\} = \varnothing$,
- l : T → A∪ {τ} - функція маркування, де A набір імен операцій. Припустимо, що $\tau \notin A$ і позначає приховану операцію.

Набір спрямованих дуг W також можна інтерпретувати як функцію. W: (P Ч T) ∪ (T Ч P) → {0, 1}. Приховані операції - це переходи, які не можуть бути спостережуваними. Вони використовуються для розділення зовнішньої і внутрішньої поведінки сервісу.

Web-сервіс - це множина $WS = (NameWs, Desc, Loc, URL, CS, SN)$, де:

- *NameWs* - це назва сервісу, яка використовується як її унікальний ідентифікатор,

- *Desc* - опис сервісу, що надається,

- *Loc* - це сервер, на якому знаходиться сервіс,

- *URL* - це виклик web-сервісу,

- *CS* є набор сервісів-компонентів web-сервісу. Якщо $CS = \{NameWs\}$, тоді *WS* – базовий сервіс. В іншому випадку *WS* є складовою послугою.

- *SN* - це модель сервісної мережі динамічної поведінки сервісів.

Позиція *i* є початковим маркуванням сервісу WS_i, тобто тільки *i* містить маркери. Виконання сервісу WS_i починається, коли маркер знаходиться в позиції *i* і закінчується, коли маркер досягає позиції *o*.

Побудова алгебри для моделювання взаємодії web-сервісів

Множина сервісів визначається за допомогою формальної граматики в нотации подібній нотації BN:

$$WS::=\varepsilon \mid X \mid Seq\ (WS_1, WS_2) \mid Choise\ (WS_1, WS_2) \mid Loop(\ WS) \mid$$
$$Conc\ (WS_1, WS_2) \mid\ ArbSeq\ (WS_1, WS_2) \mid WS_1 \|_C WS_2 \mid (WS_1 | WS_2) \rightarrow WS_3 \mid$$
$$[WS_1\ (p_1, q_1): WS_2\ (p_n, q_n)]$$

де:

– ε являє собою порожній сервіс, тобто сервіс, який не виконує ніяких операцій.

– *X* являє собою сервіс-константу, яка визначає *базовий* сервіс.

– $(Seq\ (WS_1, WS_2))$ – представляє собою комбінований сервіс, який виконує завдання сервісу WS_1, за яким виконується завдання сервісу WS_2. $Seq\ (WS_1, WS_2)$ є оператором *послідовності*. Якщо нема різниці якій сервіс виконати першим, а який потім, то такий порядок дій називається невпорядкованою послідовністю. На практиці порядок вирішується за будь-якою умовою, так як порядок не важливий, то неупорядковану послідовність також можна розглядати як *послідовність*.

– $(Choise\ (WS_1, WS_2))$ – представляє собою комбінований сервіс, який виконується як сервіс WS_1, або сервіс WS_2. Коли один з сервісів обрано, тоді інший сервіс не розглядається. $Choise\ (WS_1, WS_2)$ – це оператор *альтернативного вибору*. Вибір не є довільним і залежить від умов. Таким чином, умовну конструкцію з булевими варіантами можна розглядати як сервіс вибору.

– (*Loop* (*WS*) – є комбінованим сервісом, який виконує певну кількість разів сервіс *WS*. *Loop* (*WS*) є оператором *ітерації*.

– (*Conc* (WS_1, WS_2)) – це комбінований сервіс, який виконує завдання сервісів WS_1 і WS_2 незалежно та паралельно. Обидва сервіси включаються одночасно, і узагальнений сервіс очікує, поки обидва сервіси WS_1 і WS_2 не будуть завершені. *Conc* (WS_1, WS_2) є *паралельним* оператором.

– (*ArbSeq* (WS_1 ,WS_2)) – представляє собою комбінований сервіс, який виконує завдання сервісу WS_1, за яким виконується завдання сервісу WS_2. Або спочатку виконується сервіс WS_2, а потім сервіс WS_1. *ArbSeq* (WS_1, WS_2) є оператором *довільної послідовності*.

– ($WS_1 \|_C WS_2$) - є комбінованим сервісом, який виконує сервіси WS_1 та WS_2 незалежно один від одного, з можливостями комунікації над множиною *C* пар операцій, тобто $\|_C$ – є *паралельним* оператором *зв'язку*.

– (($WS_1 | WS_2$)→ WS_3) - є комбінованим сервісом, який очікує виконання одного з сервісів (або WS_1, або WS_2), перш ніж активувати наступний сервіс WS_3. → – оператор *дискримінатора*. Сервіси WS_1 та WS_2 не зв'язані між собою.

– [WS_1 (p_1, q_1) : WS_n (p_n, q_n)] – *динамічний вибір*, є комбінованим сервісом, який динамічно вибирає один сервіс серед *n* доступних сервісів WS_1, …, WS_n і виконує його. Він працює наступним чином: спочатку запит відсилається постачальнику послуг першого сервісу на їх вхідні точки доступу p_1, …, p_n. Тоді, виходячи з отриманих відповідей, з їх вихідних точок доступу q_1, …, q_n, та відповідно до певних критеріїв, обирається найкращий постачальник сервісу. Відповідно [:] є оператором динамічного вибору.

– Формальне визначення операторів взаємодії в термінах мереж Петрі наступне. Нехай WSi =(NameWsi, Desci, Loci, URLi, CSi, SNi) для SNi=(Pi, Ti, Wi, ii, oi, li) для i=1,…,n , буде n web-сервісів таким щоб Pi ∩ Pj = ∅ , та Ti ∩ Tj = ∅ для i ≠ j.

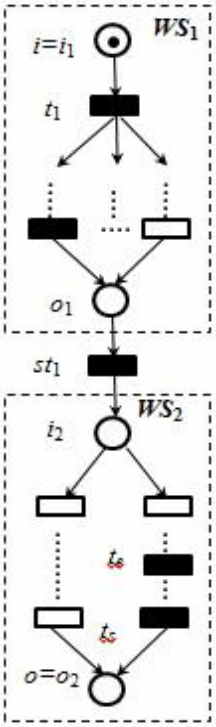
– Склад сервісів може застосовуватись до синтаксично різних складних веб-сервісів. Це пов'язано з тим, що для правильної взаємодії позицій та переходів сервісів вони не повинні перетинатися.

Базові конструкції алгебри сервісів

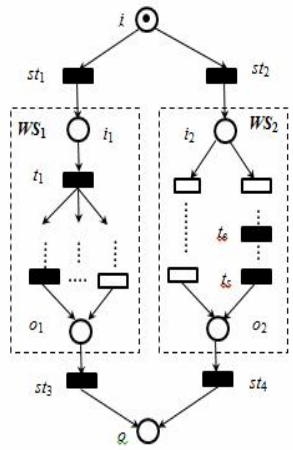
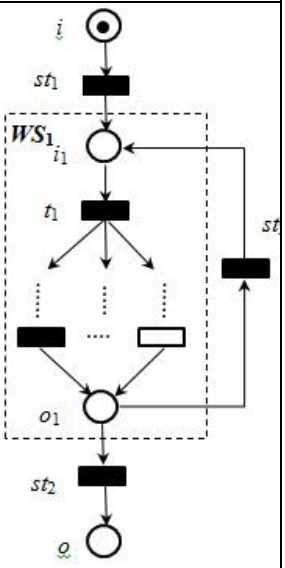
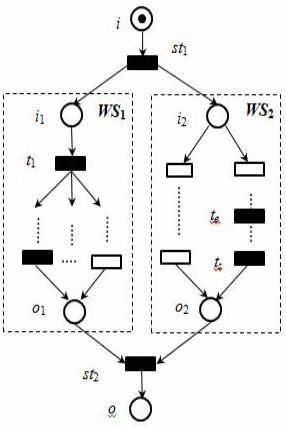
В таблиці 1 представлено базові конструкції алгебри сервісів.

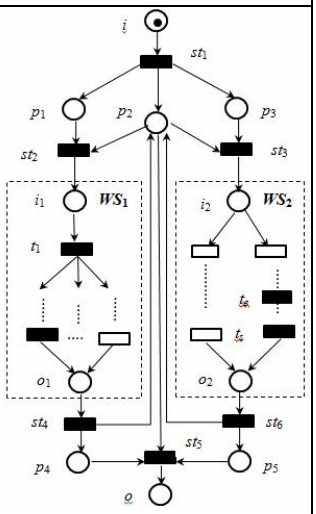
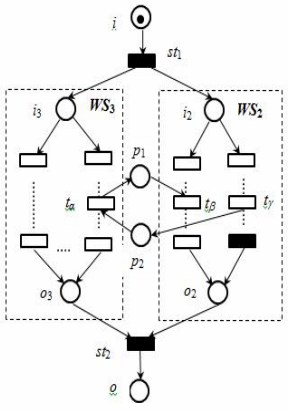
Таблиця 1

Базові конструкції алгебри сервісів

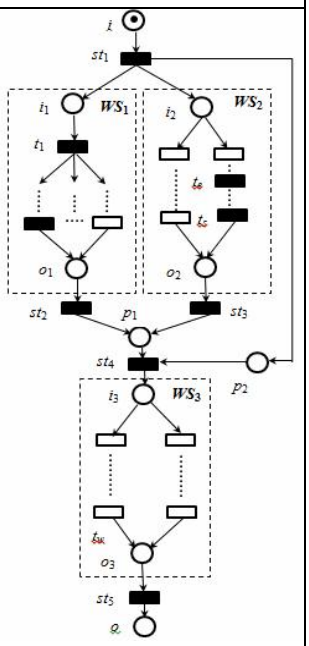
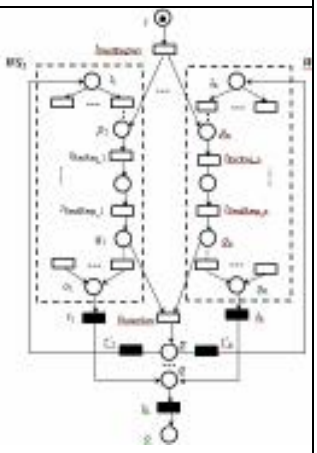
№	Позначення сервісу	Опис сервісу	Математичний опис	Графічне представлення
1.	ε	Порожній сервіс ε це сервіс, який не виконує ніяких операцій. Це умовний сервіс, він використовується для теоретичних висновків	Порожній сервіс ε визначається, як ε = (NameWs, Desc, Loc, URL, CS, SN), де: - NameWs = Порожній сервіс - Desc = «Порожній webсервіс», - Loc = Null, (нема сервера для обслуговування), - URL = Null, (нема URL-адреси ε), - CS = {∅ }, (нема сервісів компонент), - SN = ({p}, ∅ , ∅ , p, p, ∅)	
2.	Послідовність Seq(WS1, WS2)	Оператор послідовності Seq (WS1, WS2) допускає викона-ння двох сервісних операцій WS1 та WS2 послідовно, тобто, один за іншим. Web-сервіс WS1 повинен бути завершений до початку web-сервісу WS2. Зазвичай це відбувається, коли один сервіс залежить від виходу попереднього сервісу.	Seq (WS1,WS2) = (NameWs, Desc, Loc, URL, CS, SN), де: - NameWs - це ім'я нового сервісу, - Desc - опис нової послуги, - Loc - це місце розташування нового сервісу (може бути на тому ж сервері, що і один з двох компонентів сервісу). - URL - це виклик нового сервісу, - CS = WS1 ∪ WS2, - SN =(P, T, W, i,o, l), де: • P = P1 ∪ P2, • T = T1 ∪ T2 ∪ {st1}, • W = W1 ∪ W2 ∪ {(o1, st1), (st1, i2)}, • i = i1, • o = o2 , • l = l1 ∪ l2 ∪{st1, τ}.	

 ISSN 1562-9945

3.	Альтернативний вибір Choise (WS1, WS2)	Альтернативний вибір Choise (WS1, WS2), з складається з двох сервісів WS1 та WS2, але виконується або WS1, або WS2, а не обидва одночасно.	Choise (WS1, WS2)= (NameWs, Desc, Loc, URL, CS, SN), де: - NameWs - це ім'я нового сервісу, - Desc - опис нового сервісу, - Loc - це місце розташування нового сервісу, - URL - це виклик нового сервісу, - CS=WS1 ∪ WS2, - N=(P,T,W,i,o,l) де: • P = P1 U P2 U {i, o}, T = T1 U T2 U {st1, st2, st3, st4}, • W = W1 U W2 U { (i, st1), (i, st2), (st1, i1), (st2, i2), (o1, st3), (o2, st4), (st3, o), (st4, o)}, • l = l1 U l2 U {(st1, τ), (st2, τ), (st3, τ), (st4, τ)}	
4.	Цикл Loop (WS1)	Оператор циклу Loop (WS1) моделює виконання комбінованого сервісу, який виконується певну кількість разів.	Loop (WS1) = (NameWs, Desc, Loc, URL, CS, SN), де: - NameWs - це ім'я нового сервісу, - Desc - опис нової послуги, - Loc - це місце розташування нового сервісу (може бути на тому ж сервері, що і один з двох компонентів сервісу). - URL - це виклик нового сервісу, - CS = WS1, - SN = (P, T, W, i, o, l), де: • P = P1 U {i, o}, • T = T1 U {st1, st2, st3} • W = W1 U { (i, st1), (st1, i1), (o1, st2), (st2, o), (o1, st3), (st3, i1)}, • l=l1 U {(st1, τ), (st2, τ), (st3, τ)}	
5.	Паралельний Conc(WS1,WS2)	Паралельний оператор Conc (WS1, WS2) моделює викона-ння комбінованого сервісу, що одночасно виконує два сервіси WS1 та WS2. Викона-ння комбінова-ного сервісу досягне-не, коли завер-шено викона-ння	Conc (WS1, WS2) = (NameWs, Desc, Loc, URL, CS, SN), де: - NameWs - це ім'я нового сервісу, - Desc - опис нового сервісу, - Loc - це місце розташування нового сервісу, - URL - це виклик нового сервісу, - CS = WS1 ∪ WS2, - SN = (P, T, W, i, o, l) де: • P = P1 U P2 U {i, o},	

		двох сервісів WS1 та WS2.	• T = T1 U T2 U {st1, st2} • W = W1 U W2 U {(i, st1), (st1, i1), (st1, i2), (o1, st2), (o2, st2), (st2, o)}, • l = l1 U l2 U {(st1, τ),(st2, τ)}.	
6.	Довільна послідовність (ArbSeq (WS1, WS2))	Оператор довільної послідовності (ArbSeq (WS1, WS2)) моделює викона-ння комбінованого сервісу, що виконує два сервіси WS1 та WS2 у довільної послідовності. Якщо першим виконано сервіс WS1, то потім виконується сервіс WS2. А якщо першим виконано сервіс WS2, то за ним виконується сервіс WS1.	(ArbSeq (WS1,WS2)) = (NameWs, Desc, Loc, URL, CS, SN), де: NameWs – це ім'я нового сервісу, Desc - опис нової послуги, Loc - це місце розташування нового сервісу (може бути на тому ж сервері, що і один з двох компонентів сервісу). URL - це виклик нового сервісу, CS = WS1 ∪ WS2, SN = (P, T, W, i, o, l) де: • P = P1 U P2 U {i, o, p1, p2, p3, p4, p5}, • T = T1 U T2 U {st1, st2, st3, st4, st5, st6}, • W = W1 U W2 U {(i, st1), (st1, p1), (st1, p2), (st1, p3), (p1, st2), (p2, st2), (p2, st5), (p2, st3), (p3, st3), (st2, i1), (st3, i2), (o1, st4), (st4, p2), (st4, p4), (p4, st5), (o2, st6), (st6, p2), (st6, p5), (p5, st5), (st5, o)}, • l = l1 U l2 U {(st1, τ),(st2, τ),(st3, τ), (st4, τ),(st5, τ),(st6, τ) }.	
7.	Паралельний оператор з комунікацією WS1 ‖C WS2	Паралельний оператор з комунікацією WS1 ‖C WS2 моделює виконання комбінованого сервісу, що виконує одночасно два сервіси WS3 та WS2 з синхронізацією та обміном інформації.	WS1 ‖C WS2 = (NameS, Desc, Loc, URL, CS, SN), де: -C – комунікаційний елемент C = {(tα,tβ)‖(tα,tβ) ∈ T3 Ч T2 U T2 Ч T3 } -NameWs - це ім'я нового сервісу, -Desc - опис нової сервісу, -Loc - це місце розташування нового сервісу, -URL - це виклик нового сервісу, -CS = WS3 ∪ WS2, -SN = (P, T, W, i, o, l) де: • P = P3 ∪ P2 ∪{i, o} ∪ {pi / (αi,βi) ϵ C}, • T = T3 ∪ T2 ∪ {st1, st2},	

 ISSN 1562-9945

			• W = W3 ∪ W2 ∪ {(i, st1), (st1, i3), (st1, i2), (o3, st2), (o2, st2), (st2, o)} ∪ {(tαi,pi), (pi,tβi) / (tαi,tβi) ϵ C} , та • l = l3 ∪ l2 ∪ {(st1, τ), (st2, τ)}.	
8.	Дискримінатор (WS1|WS2)→WS3	Оператор дискримінаторавикористовується, для підвищення надійності web-сервісів. Він об'єднує ресурси сервісів для отримання комбінованого сервісу, який має більшу ступінь надійності сервісу	(WS1|WS2)→WS3 = (NameS, Desc, Loc, URL, CS, SN), де: -NameWs - це ім'я нового сервісу, Desc - опис нового сервісу, -Loc - це місце розташування нового сервісу, -URL - це виклик нового сервісу, -CS = WS1 ∪ WS2∪ WS3, -SN = (P, T, W, i, o, l) де: • P = P1 ∪ P2 ∪ P3 {i, o, p1, p2 }, • T = T1 ∪ T2 ∪ T3 {st1, st2, st3, st4,st5}, • W = W1 ∪ W2 ∪ W3 ∪ {(i, st1), (st1, i1), (st1, i2), (st1, p2), (o1, st2), (o2, st3), (st2, p1), (st3, p1), (p1, st4), (p2, st4), (st4, i3), (o3, st5), (st5, o)} , та • l = l1 ∪ l2 ∪ l3 ∪ {(st1, τ), (st2, τ), (st3, τ), (st4, τ),(st5, τ)}.	
9.	Динамічний вибір [WS1 (p1, q1) : WSn (pn, qn)]	Оператор динамічного вибору моделює виконання комбінованого сервісу, який дозволяє обирати найкращого постачальника сервісу серед кількох конкуруючих постачальників. Конструкція динамічного вибору дозволяє обирати кращого постачальника сервісу серед існуючих, використовуючи критерії ранжування.	[WS1 (p1, q1) : WSn (pn, qn)] = (NameS, Desc, Loc, URL, CS, SN), де: -NameWs - це ім'я нового сервісу, -Desc - опис нового сервісу, -Loc - це місце розташування нового сервісу, -URL - це виклик нового сервісу, - CS = $\bigcup_{i=1}^{n} WS_i$ - SN = (P, T, W, i, o, l) де • P = $\bigcup_{i=1}^{n} P_i \cup$ {i, o, p, q }, • T = $\bigcup_{i=1}^{n} T_i \cup$ { tSendReqServ, tSelectServ, to} ∪ {ti, ti' | 1 <i < n}, • W = $\bigcup_{i=1}^{n} W_i$ ∪ {(i, tSendReqServ), (tSelectServ, p), (q, to), (to, o)} ∪ ∪ { (tSendReqServ, pi),	

			(qi, tSelectServ), (p, tj'), (tj', ij), (oj, tj), (tj, q) | 1 <i < n}, та · l = $\bigcup_{i=1}^{n}$ ∪ {(tSendReqServ, τ), (tSelectServ, τ), (to ,τ) } ∪ {(ti, τ),(ti', τ) | 1 <i < n}.	

Кожен сервіс містить дві окремі частини, одну частину для обробки запиту на обслуговування, а інша частина – повинна виконувати сервіс самостійно. Запропонована алгебра перевіряє властівість замикання. Це гарантує, что кожен результат операции над сервісамі є сервісом, до якого можна знову застосувати алгебраїчні оператори. Таким чином, можливо будуваті більш складні та комбіновані сервіси шляхом агрегування и повторного використання існуючіх сервісів за допомогою декларатівних виразів сервісної алгебри.

Приклад моделювання взаємодії веб-сервісів

В прикладі приведеному на рис.1 представлено рієлтерський веб-сервіс, що складається з двох основних сервісів RWS (рієлтерський сервіс) та HDWS (продавець будинків).

Після отримання запиту від клієнта, RWS запускає, паралельно, аутсорсинг послуги для замовлення у сервіса HDWS варіантів житла, виконуючи операції отримує Ord_H і відправляе Del_H відповідно. Набір елементів зв'язку: C1 = {(send_ord_H, rec_ord_H), (send_del_H, rec_del_H)} и C2 = {(send_ord_H, rec_ord_H), (send_del_H, rec_del_H)}. Як тільки варіанти будуть отримані, RWS виконує агрегацію інформації.

Припустимо тепер, що замість базового сервісу HDWS, ми використовуємо композитний сервіс HD. Варіанти житла можуть бути динамічно обрано серед доступних послуг (наприклад, WS1,...WSn), які є варіантами житла.

Висновки

Семантика представленої алгебри може бути використана для доказу алгебраїчних властивостей конструкцій web-сервісів. Це може бути тоді, коли створюється комплекс web-сервісів, на основі об'єднання існуючих web-сервісів з використанням операторів алгебри. Алгебраїчні властивості можуть потім використовуватись для перетворення і оптимізації комбінованих web-сервісів на основі наступних операційних показників web-сервісів, таких як вартість і тривалість

безвідмовної роботи. Використання сервісної алгебри з представленим набором операцій дозволяє використовувати її для моделювання динамічної взаємодії веб-сервісів за допомогою кольорових мережі Петрі.

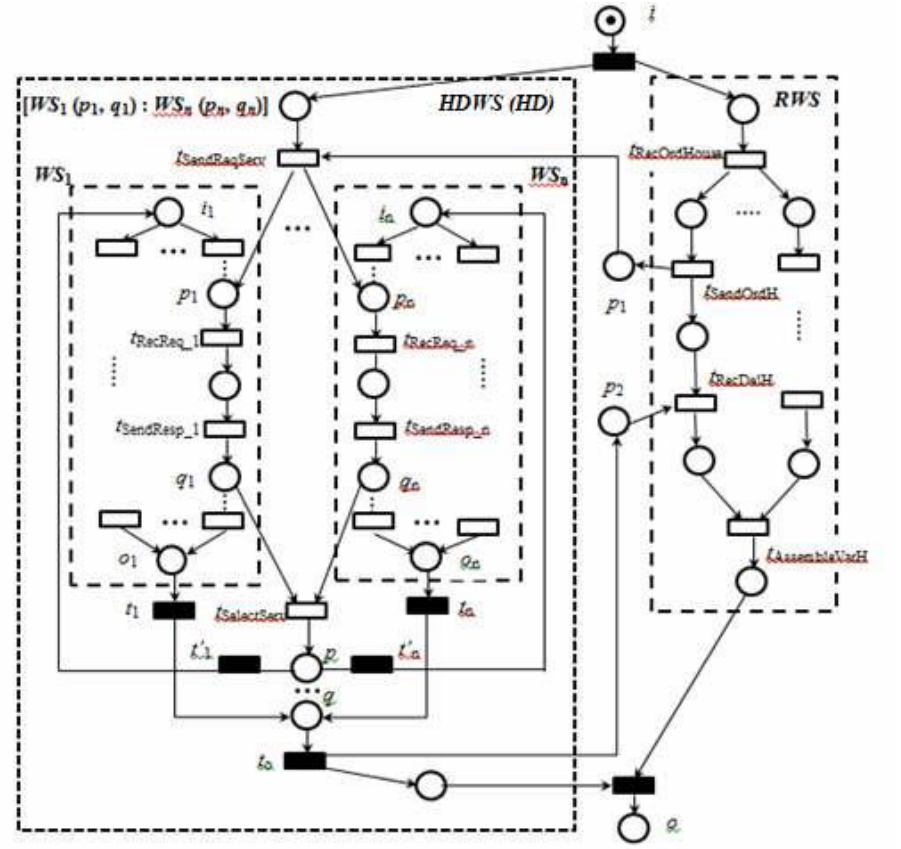


Рисунок 1 - Рієлтерський веб-сервіс

ЛІТЕРАТУРА

1. Алиев Т.И. Основы моделирования дискретных систем / Т.И. Алиев. – СПб: СПбГУ ИТМО, 2009. – 363 с.
2. Бодянский Е.В., Кучеренко Е.И., Михалев А.И. Нейро-фаззи сети Петри в задачах моделирования сложных систем : монография. Днепропетровск : Системные технологии, 2005. 311 с.
3. Бохан К.А. Модели корпоративных сервисов на основе иерархических сетей Петри / К.А. Бохан, М.С. Худолей // Радиоэлектронные и компьютерные системы. – 2010. – Вып. 47. – С. 36-41.
4. Котов, В.Е. Сети Петри [Текст] / В.Е. Котов. – М.: Наука. Главная редакция физико– математической литературы, 1984. – 160 с.
5. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. / Дж. Питерсон. – М.: Мир, 1984. – 264 с.

6. Bause, F. Stochastic Petri nets: an introduction to the theory [Text] / F. Bause, Pieter S. Kritzinger. – Friedrich Vieweg & Sohn Verlag, 2002. – 223 p.
7. Hamadi R. and Benatallah B. "A Petri net based-model for web service composition", in proc. the 14th Australasian database conference, adelaide. Darlinghurst: Australian Computer Society, 2003, pp. 191-200.
8. Jensen K., Kristensen L.M., Wells L. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. Software Tools for Technology Transfer manuscript. 2007. 40 p.
9. Genrich H. J. Lautenbach K. "System modeling with high level Petri nets", Theorical Computer Science, Vol. 13, pp. 109-136, 1981.
10. Modelling with Generalized Stochastic Petri Nets [Text] / M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis. – John Wiley & Sons, 1995. – 324 p.
11. Murata T. "Petri nets: Properties, analysis and applications," in Proc. of the IEEE, Vol. 77(4), 1989, pp. 541580.
12. Perkusich J. and J. C. A. De Figueiredo, "G-nets: A Petri net based approach for logical and timing analysis of complex software systems", Journal of Systems and Software, Vol. 39, Issue 1, pp. 39-59, Oct 1997.
13. Zaitsev D.A. Simulating Telecommunication Systems with CPN Tools: Students' book / D.A. Zaitsev, T.R. Shmeleva. – Odessa: ONAT, 2006. – 60 p.

 ISSN 1562-9945