

П.І. Андон, О.О. Слабоспицька

## ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ТА ЗАСВІДЧЕННЯ ЯКОСТІ КОНТЕКСТНО-ЗАЛЕЖНОЇ КОМПОЗИЦІЇ СЕМАНТИЧНИХ ВЕБ-СЕРВІСІВ

Розроблений авторами метод динамічної композиції адаптивного семантичного Веб-сервісу розвинуто алгоритмами узгодженого застосування спеціальної OWL-S-специфікації контексту його виконання на етапах життєвого циклу та опрацювання циклічних залежностей на функціональному рівні композиції за допомогою формалізмів Сервісу-вузла, Сервісу-посередника й Спрощення – для сталого забезпечення якості формованого Веб-сервісу. Запропоновано засвідчення рівня якості для причетних сторін шляхом динамічної верифікації, зокрема, перевірки відповідності запитаному контексту та живості, з використанням процесного числення контекстно-залежних амбієнтів. Надані алгоритми підвищують відповідність формованого Веб-сервісу очікуванням запитувачів та уможливають його стале контекстно-залежне уточнення для уніфікованої підтримки як змінних розподілених ділових процесів сучасних організацій, так і потреб споживачів певної галузі.

Ключові слова: OWL-S, семантичний Веб-сервіс, динамічна контекстно-залежна композиція Веб-сервісів, циклічна залежність, засвідчення якості, граф композиції, числення амбієнтів

### Постановка проблеми

Для ефективної автоматизованої підтримки (не)передбачено змінних розподілених ділових процесів сучасних організацій в актуальних наразі слабо формалізованих предметних областях (ПрО), де ресурси її створення дедалі обмеженіші, вимоги до якості – жорсткіші, а очікування суб'єктів і параметри інформаційної інфраструктури істотно різномірні, – все більш запитаними стають (само)адаптивні сервіс-орієнтовані програмні системи (СоПС). Вони являють собою on-line композиції (семантичних) Веб-сервісів, здатні до автоматизованого змінення поведінки для задоволення нових вимог і пристосування до нових (не)передбачених ситуацій під час виконання [1].

Призначення СоПС і властива їм життєвому циклу (ЖЦ) глибока невизначеність [2] зумовлюють актуальний виклик їх конструювання – розроблення засобів не лише сталого забезпечення, але й засвідчення рівня якості, прийняттого для всіх причетних сторін, тобто збирання, аналізу, синтезу й адресного поширення вірогідних свідчень, що СоПС задовольняє поточні функційні й нефункційні вимоги в її ЖЦ [2, 3]. Аналіз актуального доробку інженерії (само)адаптивних СоПС [3] висвітлює перспективний підхід до опрацювання цього

виклику – адекватну онтологічно базовану формалізацію контексту виконання СоПС та її складників й узгоджене застосування формалізованого контексту на стадіях ЖЦ для підвищення рівня якості їх продуктів.

Стаття підсумовує результати досліджень проекту ДР 0112U002764 під керівництвом академіка НАН України, д.ф.-м.н. П.І.Андона з розвитку, в руслі зазначеного підходу, нового методу побудови адаптивного композитного семантичного Веб-сервісу (АКС) [4,5], запропонованого авторами на підставі аналізу обмежень показових підходів до динамічної композиції семантичних Веб-сервісів для уніфікованої підтримки як розподілених і змінних процесів сучасних організацій, так і різномірних змінних потреб окремих споживачів певної ПрО.

### Формальні засоби розвитку методу

На підставі зіставлення перспективних методів контекстно-залежної композиції семантичних Веб-сервісів, поданих у табл.1, для розвитку методу [4,5] вибрано:

а) доповнення моделі АКС [4] повноцінним ядром індустріально апробованих методів on-line відшукування й компонування (семантичних) Веб-сервісів на функціональному рівні (на ґрунті методів

Таблиця 1.  
Перспективні методи контекстно-залежної композиції семантичних Веб-сервісів

№	Метод (назва, джерело)	Урахування контексту			Засоби	
		опис	етапи	техніки	специфікації	верифікації
1	Динамічна покрокова композиція А.Буччіароне [13]	Доменні об'єкти	Визначення композиції	Агрегування систем з переходом між станами	Агрегування систем з переходом між станами	Планування шляхом перевірки моделей
2	Узагальнена композиція семантичних Веб-сервісів С.Бансал [7]	Перед- і пост-умови, побічні впливи з їх об'єктами	Визначення композиції	OWL-S	Відсутні	Відсутні
3	Верифікована графо-подібна композиція семантичних Веб-сервісів для запитаного контексту виконання Р.Бен Ламін [8-10]	Онтологія контексту [9]	Всі етапи ЖЦ	Розширення мови OWL-S Відбір компонентних сервісів ССА [9]	Числення контекстно-залежних амбієнтів (ССА) [8,9]	Середовище виконання ССА [8]
4	Ціле-кероване планування Л.Ю [14]	Онтологія контексту	Опис контексту Визначення композиції	Долучення концептів контексту Відбирання Веб-сервісів	Відсутні	Відсутні
5	Аспектне програмування Л.Лі [15]	Онтологія контексту	Виконання композиції	Зв'язування аспектів	Відсутні	Відсутні
6	MDA/Notify, Decide, Configure Х.Байдури [16]	UML-модель	Визначення композиції	Погляди на кон-текст	Відсутні	Відсутні
7	Планування/семантичне зіставлення вхідних і вихідних концептів Х.Мчейк [17]	Розширення мови OWL-S	Опис контексту Відшукування Веб-сервісів Виконання композиції	Розширення OWL-S Відбір Веб-сервісів Зв'язування аспектів	Відсутні	Відсутні
8	Планування/застосування мови PDDL А.Фурно [18]	Онтологія OWL-SC	Опис контексту Визначення композиції	Розширення OWL-S Адаптаційні правила	Відсутні	Відсутні
9	Графо-орієнтоване планування Ю.Ян [19]	Показники QoS	Опис контексту Визначення композиції	Долучення QoS Відбирання Веб-сервісів	Відсутні	Відсутні
10	Застосування високо-рівневих мереж Петрі К.Букаді [20]	Мова XML	Визначення композиції	Погляди на контекст	Кольорові мережі Петрі	Планування шляхом перевірки моделей
11	Г.Алфerez, В.Пелечано [11]	Онтології Семантичного Вебу	Те ж саме	Моделювання за допомогою RDF і даних спеціальних типів	ВРМN, динамічна модель властивостей	Те ж саме
12	Трифазний підхід з використанням нечітких множин М.Мадкур [21]	Нечіткі множини	---	Правила адаптування політик застосування сервісів	Відсутні	Відсутні

М.Родрігеса [6], С.Бансал [7], Р.Бен-Ламіна [8-10], Г.Алфереца [11]) та на процесному рівні (М.Пісторе [12]);

б) удосконалення процедури композиції АКС алгоритмами опрацювання циклічної залежності на функціональному рівні за допомогою побудови сервіса-вузла [22], а в разі його неефективності – *сервіса-посередника* й *спрощення* згідно із запропонованими визначеннями;

в) засвідчення якості АКС за допомогою процесного числення контекстно-залежних амбієнтів (Ф.Сів) [23].

Долучення до ядра методів Р.Бен-Ламіна та Г.Алфереца для запитаного контексту виконання зумовлено їх перевагами для забезпечення й засвідчення якості АКС. Як показано в табл.1, серед схарактеризованих методів лише вони передбачають онтологічно базовану формалізацію контексту виконання композитного Веб-сервісу та його компонентів, відокремлену від опису виконуваних функцій, та її узгоджене застосування на етапах композиції. Завдяки цьому постачальник може зазначати в опублікованому описі (композитного) Веб-сервісу найдоцільніший, на його думку, контекст виконання, а споживач – компонувати й верифікувати цей сервіс для бажаного контексту.

### **Графо-подібна композиція семантичних Веб-сервісів**

#### **для заданого контексту виконання**

**Засади методу.** Відповідність формованого композитного Веб-сервісу заданому контексту його виконання забезпечують п'ять взаємопов'язаних тез [9,10]:

а) онтологічно-базована формалізація контексту виконання семантичних Веб-сервісів та їх композиції, уніфікована щодо етапів їх ЖЦ і ролі суб'єкта контексту (постачальник, брокер, споживач тощо);

б) узгоджене використання формалізованого контексту на всіх етапах ЖЦ композитного сервісу: від формування вимог до його публікування і/або виконання;

в) кількісне врахування, в планувальному графі композиції, трьох можливих ступенів семантичної відповідності вхідних і вихідних концептів сервісів у його складі (Exact, Plugin, Subsume) й доповнення гра-

фа таблицею індексів сумісності контекстів цих сервісів із запитаним контекстом;

г) вилучення з отриманого планувального графа всіх підграфів, відповідних запиту на композицію, та їх упорядкування за спаданням оцінок задовільності для запитувача – суми оцінок семантичної узгодженості й контекстної сумісності, зважених згідно з його перевагами;

д) використання спеціального процесного числення – Контекстно-залежних Амбієнтів (Context-aware Ambients, CCA) [23, 24] для формальної специфікації й верифікації композитного Веб-сервісу, насамперед перевірки його властивостей відповідності контексту запитувача та уточненої живості – «композитний сервіс завжди надає запитаний користувачем результат».

Наведені тези свідчать, що розглядуваний метод являє собою природний розвиток базового методу композиції на функціональному рівні М.Родрігеса [6] і є доцільним для застосування після нього в процесі покрокової композиції АКС [5].

### **Моделі та алгоритми композиції.**

Для реалізації тез а) – д) передбачено дві групи формалізмів [9,10]:

а) моделі та відповідні їм машинно-читані описи:

контексту виконання семантичного Веб-сервісу [25,26];

семантичного Веб-сервісу в заданому контексті [27];

запиту споживача щодо композиції семантичних Веб-сервісів для заданого контексту;

композитного семантичного сервісу, формованого за запитом;

б) алгоритми формування й застосування моделей:

прямого пошуку для побудови повного прошарованого графа компонентних Веб-сервісів згідно із запитом на їхню композицію;

зворотного пошуку для вилучення всіх нескорочуваних підграфів, відповідних запиту, які й являють собою моделі пошукуваних композитних сервісів;

впорядкування отриманих композитних сервісів за спаданням рейтингів їх прийнятності для запитувача, обчислених як сума індексів семантичної відпо-

відності між компонентними сервісами та відповідності їх контекстів запитаному контексту, зважених згідно з перевагами запитувача [25].

У межах методу контекст охоплює «довільні відомості, які можна використати для опису ситуації щодо сутності...– особи, місця або об'єкта, пов'язаного із взаємодією користувача й застосунку, включно із самими користувачем і застосунком» [8]. *Модель контексту* являє собою підмножину концептів спеціальної онтології контекстів [25,26] із їх взаємозв'язками. Ця онтологія є багаторівневим розширенням онтології OWL-S для уніфікованого опису множин контекстів у довільних ПрО мовою OWL. На її верхньому рівні відображено основні відомості щодо контексту за допомогою фіксованої структури концептів, наведеної на рис.1. Докладніші відомості визначено на (поповнюваних) нижчих рівнях, залежних від ПрО виконання композитного Веб-сервісу.

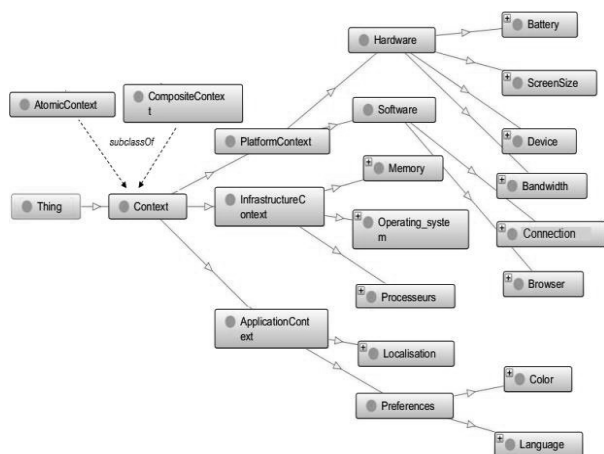


Рис. 1. Структура верхнього рівня онтології контекстів

Як показано на рис. 1, концептами верхнього рівня є «Контекст застосунку», «Платформний контекст», «Інфраструктурний контекст», а також «Атомарний контекст» і «Складений контекст». Запровадження останнього концепту уможливує подання контексту композитного сервісу як перетину атомарних контекстів його складників для точнішого відображення очікувань запитувачів і ситуацій у середовищі виконання.

Опис семантичного Веб-сервісу, виконаного в заданому контексті, являє со-

бою де-факто стандартну трійку Д.Мартіна – профіль, модель та обґрунтування сервісу [27], де, однак, згідно з пропозиціями [28] профіль сервісу доповнено атрибутом «adaptedTO» («пристосований\_ДО»). Значенням цього атрибута є URL зовнішнього файлу з описом динамічного контексту, найдоцільнішого для виконання сервісу, на думку його постачальника. Приклад такого опису з атрибутом «adaptedTO» для профілю Веб-сервісу огляду ресурсів Інтернет подано на рис. 2.

Для зберігання доповнених описів передбачено відповідний розвиток реєстру UDDI [29], ефективність якого засвідчує його макетна реалізація з використанням JUDDIv3 і бази даних MySQL [8].

Запропоновані описи контексту та семантичного Веб-сервісу в заданому контексті дозволяють постачальнику сервісу легко оновлювати нефункціональні відомості щодо контексту в зовнішньому файлі без змін самого опису. Вони також уможливають формальну постановку проблеми композиції семантичних Веб-сервісів для заданого контексту за допомогою

*Визначення 1* [9]. Модель семантичного Веб-сервісу із заданим контекстом – це трійка

$$w = \langle I, O, C \rangle, \quad (1)$$

де  $I$  й  $O$  – множини його вхідних і вихідних параметрів (концептів онтології певної ПрО);

$C$  – множина параметрів контексту (концептів онтології контекстів).

*Визначення 3.2* [9]. Запит на композицію семантичних Веб-сервісів для заданого контексту – це п'ятірка

$$rc = \langle In, Out, Cr; ws, wc \rangle, \quad (2)$$

де  $In$  і  $Out$  – множини вхідних і вихідних запитаних параметрів,

$Cr$  – множина параметрів контексту користувача;

$ws$  і  $wc$  – вагові коефіцієнти відносно важливості для користувача семантичної узгодженості формованого композитного сервісу та його відповідності запитаному контексту.

*Визначення 3.3* [9]. Постановка проблеми композиції семантичних Веб-сервісів для заданого контексту – це структурований кортеж

```

<profile:Profile rdf:ID="__DESTINATION_PROFILE">
<service:isPresentedBy rdf:resource="#_DESTINATION_SERVICE"/>
<profile:serviceName xml:lang="en">
A_MyDESTINATIONService
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns DESTINATION of the organization responsible for providing surfing activities.
</profile:textDescription>
<profile:hasInput rdf:resource="#_ORGANIZATION"/>
<profile:hasInput rdf:resource="#_SURFING"/>
<profile:hasOutput rdf:resource="#_DESTINATION"/>
<profile:has_process rdf:resource=" DESTINATION_PROCESS" />
<profile:adaptedTo
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
http://localhost/services/AOWLS/contexts/Context\_destination\_MyOfficeservice.xml
</profile:adaptedTo>
</profile:Profile>

```

Рис. 2. Приклад опису Веб-сервісу огляду ресурсів Інтернет

$cp = \langle O, R, CO, rc; (\langle cs_i(rc), s_i, ss_i, sc_i \rangle, i \geq 1) \rangle$ , (3)

$s_i = (ws \times ss_i + wc \times sc_i) / (ws + wc)$ ,

де  $O$  – задана онтологія ПрО, для якої запитано композицію;

$R$  – множина семантичних Веб-сервісів для заданого контексту з моделлю (1) (об'єднання вмісту доступних реєстрів Інтернет);

$CO$  – онтологія контекстів;

$rc$  – запит на композицію (2);

$cs_i(rc)$  –  $i$ -й композитний семантичний сервіс для запиту  $rc$  в їх множині, впорядкованій за спаданням оцінок їх задовільності для споживача  $s_i$ ;

$ss_i$  й  $sc_i$  – оцінки семантичної узгодженості формованого композитного сервісу та його відповідності запитаному контексту [25], розглянуті далі.

Алгоритми вирішення поставленої проблеми (3) передбачають побудову на множині  $R$  повного планувального графа в його традиційному форматі [10], який, однак, стає контекстно-залежним, оскільки його вершинами є сервіси з певними контекстами (1). Під час його формування враховують усі можливі ступені семантичної подібності між вихідним ( $out$ ) і вхідним ( $in$ ) концептами послідовно відшукуваних проміжних сервісів, яким додатково зіставлені кількісні оцінки  $e(out, in)$  [10]:

– 1, якщо об'єми відповідних концептів однакові, тобто ступінь подібності – Exact;

– 0.9, якщо вихідний концепт одного сервісу є вкладеним концептом вхідного концепту іншого, тобто ступінь подібності – Plugin;

– 0.5, якщо для вихідного концепту одного сервісу вхідний концепт іншого є вкладеним концептом, тобто ступінь подібності – Subsume;

– 0, якщо семантична подібність між вихідним і вхідним концептами відсутня (ступінь Fail).

На підставі цих оцінок формується допоміжна матриця семантичних зв'язків, вигляд якої фіксує технічне

*Визначення 4* [10] Матриця семантичних зв'язків для постановки проблеми композиції (3) – квадратна матриця з елементами  $sl_{ij} = 0$ , якщо серед вихідних концептів  $i$ -го сервісу  $w_i \in R$  і вхідних концептів  $j$ -го сервісу  $w_j \in R$  немає семантично подібних;

$sl_{ij} = (V_{ij}; sim_{ij})$ ,  $V_{ij} = \{(out, in), out \in O(w_i), in \in I(w_j)\}$ ; (4)

– в іншому випадку, причому  $sim_{ij}$  – середнє оцінок  $e(out, in)$  для множини  $V_{ij}$ .

Процедура вирішення проблеми композиції в постановці (3) передбачає три обов'язкові кроки:

а) побудова повного контекстно-залежного планувального графа та відповідних йому матриць індексів контекстної подібності ( $CS$ ) й семантичних зв'язків ( $SLM$ ) за допомогою алгоритму прямого пошуку, поданого на рис. 3;

б) вилучення з побудованого графа відповідних запиту підграфів  $cs_i(rc)$  з (3) та обчислення оцінок задовільності відповідних композитних семантичних Веб-сервісів для споживача  $s_i$  із складниками  $ss_i$  й  $sc_i$  за допомогою алгоритму зворотного пошуку;

в) впорядкування отриманих композитних сервісів за спаданням оцінок  $s_i$  за допомогою алгоритму сортування.

Четвертим необов'язковим кроком є публікування композитних сервісів з

```

Input : ReqComp < InReq, OutReq, CReq >: User's Request
        R : Repository of Adaptable Semantic Web Services
        O : Domain Ontology
        CO : Context Ontology
Output: G : Context Aware Semantic Plannin Graph
1 A0=∅; P0={InReq}; G=(A0,P0)
2 i=1;
3 Init_SLM(SLM);
4 repeat
5   S=Select_SimSC_Services(Pi-1, R, O, CO);
6   Pi=Pi-1;
7   for each s in S do
8     if (! Exist(s)) then
9       SLM=Update_SLM;
10      Add(Pi, s.out);
11      CS=Build_CS(CS,s);
12      Ai=Ai ∪ {s};
13    end
14  end
15  G=G ∪ {(Ai,Pi)};
16  i=i+1;
17 until OutReq ⊆ Last_Pi(G) ∨ Fixed_Point(G);
    
```

Рис. 3. Псевдокод алгоритму прямого пошуку найкращою оцінкою  $s_i$  у розширенні реєстру UDDI [29].

Як показано на рис. 3, застосування всіх можливих ступенів семантичної подібності вхідних і вихідних концептів для формування ярусів планувального графу підвищує кількість його підграфів, відповідних запиту, оскільки повний збіг таких концептів може не досягатися.

На початку виконання алгоритму планувальний граф містить шар 0, де множина дій  $A_0$  порожня, а множина висловлювань  $P_0$  подає початковий стан проблеми планування й містить вхідні параметри запиту. В досліджуваному випадку композиції семантичних Веб-сервісів кожна дія в  $G$  подає атомарний Веб-сервіс. Вхідні параметри Веб-сервісу стають передумовами дії в графі

$G$ , а вихідні параметри відображаються в позитивні наслідки дії.

Планувальний граф  $G$  ітеративно поповнюють, доки не буде виконано одну з умов завершення алгоритму:

- досягнуто рівня, для якого множина пропозицій містить усі вихідні параметри з запиту на композицію (2);
- досягнуто фіксованого рівня  $i$ , де  $A_{i+1} = A_i$ ,  $P_{i+1} = P_i$ .

Для кожного шару  $i > 0$  нові Веб-сервіси відшуковують в репозиторії  $R$  на підставі семантичного зіставлення їхніх вхідних параметрів з висловлюваннями  $(i-1)$ -го шару та сумісності контекстів, виявлених Веб-сервісів із запитаним контекстом користувача (*Select\_SimSC\_Services* на рис. 3). Для кожного нового відшуканого Веб-сервісу, ще не долученого до графу  $G$ , додають нові елементи до матриць семантичних зв'язків (*Update\_SLM*) та індексів контекстної подібності (*Build\_CS*), а його вихідні параметри долучають до множини висловлювань  $P_i$ .

Побудова планувального графа з матрицями  $CS$  і  $SLM$  уможливорює формування множини найкращих рішень проблеми композиції (3) (*ACSWS\_List*) за допомогою алгоритму зворотного пошуку, поданого на рис. 4.

Згідно з рис. 4, пошук починають з останнього шару висловлювань у планувальному графі  $G$ .

Нехай  $g$  – множина цільових висловлювань, яких необхідно «досягти» для заданого шару висловлювань  $P_i$ . Алгоритм знайде підмножину дій з  $A_i$ , які разом «досягають»  $g$ , і відбере ті з них, що мають найвище значення семантичної узгодженості, на підставі матриці семантичних зв'язків з оцінками (4). Об'єднання передумов цих дій являє собою нову множину цілей, яких треба досягти для попереднього шару висловлювань  $P_{i-1}$ . Виконання цього процесу за алгоритмом, наведеним на рис. 5, продовжують, поки не буде досягнуто першого шару висловлювань графа  $G$ , а саме  $P_0$ .

На кроці впорядкування алгоритм сортування (див. рис. 6) приймає множину найкращих композитних Веб-сервісів, сформовану на попередньому кроці, й надає їх перелік *Orders\_Sols*, впорядкований за спаданням оцінок (4).

**Input** :  $ReqComp < In_{Req}, Out_{Req}, C_{Req} >$ : User's Request  
 $G$ : Context Aware Semantic Plannin Graph  
 $SLM$ : Semantic Link Matrix  
 $CS$ : Context Similarity Table

**Output**:  $ACSWS\_List$  :  
 List of best adaptable composed Semantic Web Services

```

1 if (!  $Out_{Req} \subseteq Last\_P_i(G)$ ) then
2   | return  $\emptyset$ ;
3 else
4   |  $ACSWS\_List := extract\_best\_sols(G, SLM, CS)$ 
5 end
```

Рис.4. Псевдокод алгоритму зворотного пошуку

**Input** :  $goals$  : set of goals to be achieved  
 $i$ : a level in the Graph  $G$   
 $G$ : Context Aware Semantic Plannin Graph  
 $SLM$ : Semantic Link Matrix  
 $CS$ : Context Similarity Table

**Output**:  $Sols$ : set of best solutions

```

1 if (  $i = 0$ ) then
2   | return  $Sols$ ;
3 else
4   |  $goals' = \emptyset$ ;
5   | for each proposition  $p$  in  $goals$  do
6     |  $sources = Get\_Sources(p)$ 
7     | /*return the set of services which semantically produce  $p$  */
8     | Choose an action  $a$  such that  $\exists source$  in  $sources$  ,
9     |  $SLM(a, source) = \max_j (SLM(a, source_j)), source_j \in sources$ 
10    |  $goals' = goals' \cup \{a.Ins\}$ 
11    |  $sol = sol \cup \{a\}$ 
12  end
13   $Sols = Sols \cup \{sol\}$ 
14   $extract\_best\_Solutions(G, SLM, goals', i - 1, CS)$ 
15 end
```

Рис. 5. Псевдокод алгоритму вилучення найкращих композитних сервісів

**Input** :  $ACSWS\_List$  : List of valid adaptable composed Semantic Web Services

**Output**:  $Ordered\_Sols$ : ranked composition solutions

```

1 for each  $sol$  in  $ACSWS\_List$  do
2   |  $Score_{sol} = [w_{sem} * score_{sem}(sol) + w_{context} * score_{context}(sol)] / (w_{sem} + w_{context})$ 
3 end
4  $Ordered\_Sols = Sort(ACSWS\_List, Score_{sol})$ 
```

Рис. 6. Псевдокод алгоритму сортування

**Специфікація й верифікація АКС із контекстом за допомогою числення контекстно-залежних амбієнтів.** Універсальний формальний апарат специфікації АКС із контекстом (2) і запиту щодо нього (3) надають темпоральні логіки лінійного й розгалуженого часу, а верифікації відповідних описів – методи й інструментальні засоби перевірки моделей за допомогою цих логік [30]. Однак вони зорієнтовані переважно на опис ділового процесу, виконуваного АКС, і залишають акцентований методом контекст

його перебігу поза увагою. Тому для за-свідчення якості формованого АКС з контекстом пропонується його специфікація та верифікації насамперед за допомогою числення контекстно-залежних амбієнтів (calculus of context-aware ambients, CCA) [23,24,31].

Числення CCA розвинуто Ф.Сівом на ґрунті відомого числення *мобільних амбієнтів* Л.Карделлі–А.Гордона для моделювання контекстно-залежних динамічно взаємодіючих систем й виведення логічних висновків щодо їхньої поведінки. Базове поняття *амбієнта* [23,24] являє собою концептуальне подання довільної сутності (фізичної, віртуальної, мобільної або непорушної), яка визначає обмежене місце, де має місце обчислення, та:

- має ім'я, межі й може бути вкладена до іншого амбієнта;
- може бути мобільною;
- може обмінюватися повідомленнями з собою та з іншими амбієнтами;
- може усвідомлювати присутність інших амбієнтів у її середовищі.

Хоча з теоретичної точки зору CCA можна природним чином «занурити» до системи інсерційного моделювання О.А.Летичевського, перспективність його застосування для побудови контекстно-залежного АКС [4,5] зумовлена наявністю відповідної йому мови програмування ссаPL і вільно доступного середовища виконання програм цією мовою [23,31].

У CCA передбачено чотири основні синтаксичні категорії [23,24], зіставлені на рис. 7:

- 1) *процеси*, які подають обчислення, тобто поведінку модельованої системи;
- 2) *спроможності* (capabilities) – елементарні дії, які можуть виконувати процеси;
- 3) *розміщення* (locations) – позиції в ієрархії амбієнтів;
- 4) *контекстні вирази* – формули запровадженої просторової логіки, що подають властивості контексту процесів.

Для CCA-верифікації АКС з контекстом у [29] виокремлено вісім патернів структур керування в сервісній моделі зі складу його опису мовою AOWL-S, яким зіставлено CCA-подання.

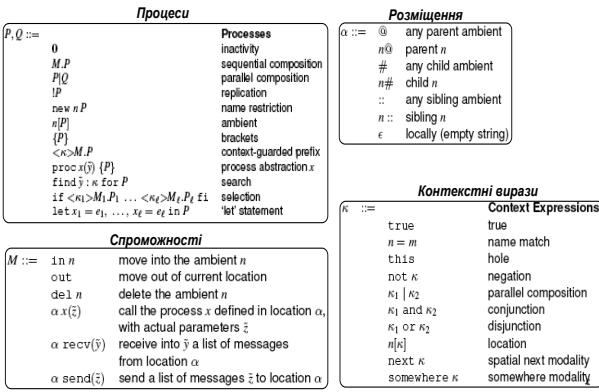


Рис. 7. Синтаксис числення ССА

1. Атомарному процесу, що реалізується атомарним семантичним сервісом з контекстом  $S(C, P)$  (де  $C$  – найдоцільніший контекст виконання  $S$ , а  $P$  – атомарний процес, який слід виконати), в ССА відповідає амбієнт з іменем  $S$ :  $S[C|P]$  (нотація  $C|P$  вказує, що процес  $P$  виконується в контексті  $C$ ).

2. Щоб моделювати послідовність композитних семантичних Веб-сервісів з контекстом, дії, охоплені послідовністю, подають амбієнтами, а відношення передування – повідомленнями, які надсилають ці амбієнти. Враховуючи ієрархічну структуру амбієнтів ССА, послідовність можна моделювати амбієнтом-батьком, що подає композитний Веб-сервіс, з амбієнтами-нащадками одного рівня  $A, B$ . У синтаксисі ССА подання послідовності має вигляд системи формальних висловлювань:

$$P_A = S_{Seq} \uparrow(x). \tau_A . B :: \langle y \rangle . 0$$

$$P_B = A :: (y). \tau_B . S_{Seq} \uparrow(z). 0$$

$$P_{Seq} = EC(C_A, C_B) ? P_A . P_B$$

або фрагмента коду мовою ссаPL:

```
<Sequence>
<Process> C_A, A </Process>
<Process> C_B, B </Process>
</Sequence>
```

Процес  $P_A$  отримує повідомлення  $x$  від амбієнта-батька (позначеного  $S_{Seq} \uparrow$ ), виконує внутрішню дію  $\tau_A$ , передає результат у своєму однорівневому амбієнту  $B$  (позначеному  $B ::$ ) і завершується. Процес  $P_B$  отримує повідомлення  $y$  від свого однорівневого амбієнта  $A$  (позначеного  $A ::$ ), виконує внутрішню дію  $\tau_B$ , надсилає результат  $z$  амбієнту-батьку  $S_{Seq}$  і завершується. Результуючий процес  $P_{Seq}$  є контекстно-захисним,

де  $EC(C_A, C_B)$  є контекстним виразом щодо найприйнятніших контекстів  $C_A$  і  $C_B$  для процесів  $P_A$  і  $P_B$ .

3. Дії, охоплені структурою керування «Розгалуження» мають виконуватися одночасно. Щоб моделювати розгалуження композитних семантичних Веб-сервісів з контекстом, попередні дії й дії, охоплені розгалуженням, слід перетворити в амбієнти ССА й застосувати оператор паралельної композиції «|» для подання відношення одночасного виконання.

У синтаксисі ССА розгалуження з діями  $B, C$ , яким передують дії  $A$ , подають таким чином:

$$P_A = S_{Split} \uparrow(x). CE(C_A) ? \tau_A . CE(C_A, C_B) ? (B :: \langle y \rangle | D :: \langle z \rangle) . 0$$

$$P_B = P_A \uparrow(y). CE(C_A, C_B) ? \tau_B . S_{Split} \uparrow \langle y_1 \rangle . 0$$

$$P_D = P_A \uparrow(z). CE(C_A, C_D) ? \tau_D . S_{Split} \uparrow \langle z_1 \rangle . 0$$

а мовою ссаPL – так:

```
<Process> C_A, A </Process>
<Split>
  <Process> C_B, B </Process>
  <Process> C_D, D </Process>
</Split>
```

Процес  $P_A$  отримує повідомлення  $x$  від свого амбієнта-батька (позначеного  $S_{Seq} \uparrow$ ). Якщо контекст виконання підтверджує контекстний вираз  $CE(C_A)$ , процес  $P_A$  виконує внутрішню дію  $\tau_A$ , одночасно надсилає повідомлення  $y$  і  $z$  своїм однорівневим амбієнтам  $B$  і  $D$  (позначеним  $B ::$  і  $D ::$ ) й завершується.

Процеси  $P_B$  і  $P_D$  отримують повідомлення  $y$  і  $z$  від свого однорівневого амбієнта  $A$  (позначеного  $A ::$ ). Якщо контекст виконання підтверджує контекстний вираз  $CE(C_A, C_B)$ , процес  $P_B$  виконує свою внутрішню дію  $\tau_B$ , надсилає результат  $y_1$  амбієнту-батьку  $S_{Split}$  і завершується.

Одночасно, якщо контекст виконання підтверджує контекстний вираз  $CE(C_A, C_D)$ , процес  $P_D$  виконує внутрішню дію  $\tau_D$ , надсилає результат  $z_1$  амбієнту-батьку  $S_{Split}$  і завершується.

4. Структура керування «Розгалуження з об'єднанням» (Split+Join) охоплює одночасні дії, які необхідно синхронізувати. Щоб моделювати розгалуження з об'єднанням композитних семантичних Веб-сервісів з контекстом, дії, охоплені послідовністю, й наступні дії слід перетворити в амбієнти ССА. Одночасно виконання внутрішніх дій наступного процесу роз-



почнуться лише після отримання повідомлень-тригерів, надісланих процесами, що охоплені Розгалуженням з об'єднанням, якщо відповідний контекстний вираз підтверджений контекстом виконання.

У синтаксисі ССА розгалуження з об'єднанням, де послідовність містить дії  $A, B$  і наступну дію  $D$ , подають так:

$$\begin{aligned} P_A &= S_{Split-Join} \uparrow (x). \tau_A. CE(C_A, C_B, C_D) ? (B :: < y > | D :: < z >). 0 \\ P_B &= CE(C_A, C_B) ? A :: (y). \tau_B. S_{Split-Join} \uparrow < t_1 > . 0 \\ P_D &= CE(C_A, C_D) ? A :: (y). \tau_D. S_{Split-Join} \uparrow < t_2 > . 0 \end{aligned}$$

а мовою ссаPL – так:

```
<Split-Join>
  <Process> CA, A </Process>
  <Process> CB, B </Process>
</Split-Join>
<Process> CD, D </Process>
```

Процес  $P_A$  отримує повідомлення  $x$  від амбієнта-батька (позначеного  $S_{Split-Join} \uparrow$ ) й виконує внутрішню дію  $\tau_A$ . Якщо контекстний вираз  $CE(C_A, C_D)$  підтверджено контекстом виконання, процес  $P_A$  надсилає повідомлення  $z_1$  своєму однорівневому амбієнту  $D$  (позначеному  $D ::$ ) і завершується. Одночасно процес  $P_B$  отримує повідомлення  $y$  від свого амбієнта-батька ( $S_{Split-Join} \uparrow$ ) й виконує внутрішню дію  $\tau_B$ . Якщо контекстний вираз  $CE(C_B, C_D)$  підтверджено контекстом виконання, процес  $P_B$  надсилає повідомлення  $z_2$  своєму однорівневому амбієнту  $D$  ( $D ::$ ) і завершується.

$P_D$  одночасно отримує повідомлення  $z_1$  і  $z_2$  від своїх однорівневих амбієнтів  $A$  і  $B$ . Тоді процес  $P_D$  виконує внутрішню дію  $\tau_D$ , посилає результат  $t$  амбієнту-батьку  $S_{Split-Join}$  і завершується.

5. Структуру керування «Вибір» застосовують, коли треба випадковим чином вибрати для виконання точно один процес з їх множини. В синтаксисі ССА вибір між процесами  $B$  і  $D$  після виконання процесу  $A$  подають таким чином:

$$\begin{aligned} P_A &= S_{Choice} \uparrow (x). \tau_A. (CE(C_A, C_B) ? B :: < y > . 0 + CE(C_A, C_D) ? D :: < z > . 0) \\ P_B &= CE(C_A, C_B) ? A :: (y). \tau_B. S_{Choice} \uparrow < t_1 > . 0 \\ P_D &= CE(C_A, C_D) ? A :: (z). \tau_D. S_{Choice} \uparrow < t_2 > . 0 \end{aligned}$$

а мовою ссаPL – так:

```
<Process> CA, A </Process>
  <Choice>
    <Process> CB, B </Process>
    <Process> CD, D </Process>
  </Choice>
```

де  $S_{Choice}$  – амбієнт-батько.

Амбієнти  $A, B$  і  $D$  є однорівневими нащадками амбієнта  $S_{Choice}$ . Амбієнт  $A$  отримує повідомлення  $x$  від амбієнта-батька  $S_{Choice}$ , виконує внутрішню дію  $\tau_A$  й далі недетерміновано вибере одного зі своїх однорівневих амбієнтів  $B$  або  $D$  для виконання. Якщо вибрано  $B$ , і контекстний вираз  $CE(C_A, C_B)$  підтверджено його контекстом  $C_B$ , то амбієнт  $A$  надсилає повідомлення у амбієнту  $B$  й завершується. Таким же чином, якщо вибрано  $D$ , і контекстний вираз  $CE(C_A, C_D)$  підтверджено його контекстом  $C_D$ , амбієнт  $A$  надсилає повідомлення у амбієнту  $D$  й завершується.

Якщо контекстний вираз  $CE(C_A, C_B)$  (відповідно  $CE(C_A, C_D)$ ) виконано, то процес  $P_B$  (відповідно,  $P_D$ ) отримує повідомлення  $x$  (відповідно,  $y$ ) від однорівневого амбієнта  $A$ , виконує внутрішню дію  $\tau_B$  ( $\tau_D$ ), надсилає результат  $t_1$  (відповідно,  $t_2$ ) амбієнту-батьку  $S_{Choice}$  і завершується.

6. Структуру «Якщо - То - Інакше» застосовують, коли потрібно вибрати для виконання одну з двох дій згідно з умовою. Щоб моделювати цю структуру для АКС з контекстом, охоплені нею дії слід перетворити в амбієнти ССА, а умові зіставити захисні вирази, які застосовують, щоб вибрати дію для виконання. Коли структура «Якщо - То – Інакше» охоплює дії  $A$  і  $B$ , її ССА-подання має вигляд:

$$\begin{aligned} P_A &= [Cond = True] CE(C_A) ? S_{If-Then-Else} \uparrow (x). \tau_A. S_{If-Then-Else} \uparrow < y > . 0 \\ P_B &= [Cond = False] CE(C_B) ? S_{If-Then-Else} \uparrow (x). \tau_B. S_{If-Then-Else} \uparrow < y > . 0 \end{aligned}$$

де  $S_{If-Then-Else}$  – амбієнт-батько, амбієнти  $A$  і  $B$  – його нащадки,  $PA$  і  $PB$  – процеси, виконувані  $A$  і  $B$ .

Цю структуру описує ссаPL-код:

```
<If-Then-Else>
  <If-Condition> Cond </If-Condition>
  <Then>
    <Process> CA, A </Process>
  </Then>
  <Else>
    <Process> CB, B </Process>
  </Else>
</If-Then-Else>
```

Якщо умову  $Cond$  задоволено, процес  $P_A$  запускається, коли контекстний вираз  $CE(C_A)$  підтверджено його контекстом  $C_A$ .  $P_A$  отримує повідомлення  $x$  від амбієнта-батька  $S_{If-Then-Else}$ , виконує внутрішню дію  $\tau_A$ , надсилає результуюче повідомлення у

$S_{If-Then-Else}$  й завершується. Коли умову  $Cond$  не виконано, процес  $P_B$  запускається, якщо контекстний вираз  $CE(C_B)$  підтверджено контекстом  $C_B$ .

7. Структура керування «Повторювати-Поки» підтримує повторення дії, доки умову підтверджено. Щоб моделювати цю структуру для АКС з контекстом, підтримувану нею дію слід перетворити в амбієнт і застосовувати умову, щоб визначити, чи продовжувати цикл. Оператор реплікації «!» застосовують для позначення повторення тіла циклу.

Якщо ця структура охоплює одну дію  $A$ , її ССА-подання має вигляд:

$$P_A = !([Cond = True]CE(C_A)?S_{Repeat\_While} \uparrow(x).\tau_A).S_{Repeat\_While} \uparrow < y > .0,$$
 де  $S_{Repeat\_While}$  – амбієнт-батько з нащадком – амбієнтом  $A$ ,  $PA$  – процес, виконуваний  $A$ .

Наведеному поданню відповідає фрагмент коду мовою ссаPL:

```
<Repeat-While>
  <While-Condition>
    Cond
  </While-Condition>
  <While-Process>
    <Process> CA, A</Process>
  </While-Process>
</Repeat-While>
```

8. На відміну від попередньої структури, ССА-подання останньої структури «Повторювати до» («Repeat-Until для АКС з контекстом має вигляд:

$$P_A = (CE(C_A)?S_{Repeat\_Until} \uparrow(x).\tau_A.P_A.[Cond = True]).S_{Repeat\_Until} \uparrow < y > .0,$$

де збережено попередні позначки.

Якщо контекстний вираз  $CE(C_A)$  підтверджено,  $P_A$  обробляє повідомлення  $x$  від амбієнта-батька  $S_{Repeat\_Until}$  і виконує свою внутрішню дію. Процес  $P_A$  рекурсивно повторюється, доки умова  $Cond$  виконана. Коли  $Cond$  порушено, процес  $P_A$  надсилає результуюче повідомлення  $y$  до  $S_{Repeat\_While}$  й завершується.

Подання цієї структури мовою ссаPL має вигляд

```
<Repeat-Until>
  <Until-Process>
    <Process> CA, A</Process>
  </Until-Process>
  <Until-Condition>
    Cond
  </Until-Condition>
</Repeat-Until>
```

### Сервіс-посередник і спрощення для усунення циклічних залежностей в АКС з контекстом

Зіставлення змістовних описів алгоритмів усунення циклічної залежності в композиції семантичних Веб-сервісів за допомогою *сервісу-вузла* [1,22], розроблених на попередньому етапі робіт за проектом ДР 0112U002764, і найпростішого формалізованого опису композитного семантичного Веб-сервісу на функціональному рівні [1,6] дозволяє формально описати сервіс-вузол за допомогою

*Визначення 5.* Нехай  $CD = \{sc_i, i \geq 1\}$ ,  $sc_i = (In(sc_i); Out(sc_i))$  – множина описів, у сенсі [1,6], семантичних Веб-сервісів, які утворюють циклічну залежність. Опис на функціональному рівні семантичного Веб-сервісу-вузла для її усунення – це пара множин його вхідних і вихідних концептів

$ck = \langle INK; OUK \rangle,$

які є об'єднаннями подібних множин для Веб-сервісів зі складу залежності:

$INK = \cup \{In(sc_i), sc_i \in CD\};$  (5)  
 $OUK = \cup \{Out(sc_i), sc_i \in CD\}.$

Вираз (5) висвітлює ситуації доцільного й недоцільного застосування *сервіс-вузла* для опрацювання циклічної залежності. Саме, він ефективний в ідеальній ситуації, коли «непродуктивні» (тобто без внеску до вихідних концептів запиту) сервіси  $sc_i \in CD$  високонадійні, з привабливими показниками якості й низькою трудомісткістю/вартістю використання, їх кількість відносно невелика порівняно з кількістю всіх компонентів композитного Веб-сервісу, а взаємозв'язки в межах залежності відносно прості. Тоді непродуктивні сервіси надають резервування функціональних сервісів – їх своєрідне «кріплення», підвищуючи якість композитного Веб-сервісу. Однак у реальній «неідеальній» ситуації складна внутрішня структура сервісу-вузла і/або незадовільна якість сервісів у її складі можуть неприйнятно знизити якість композитного сервісу аж до його недоступності.

«Неідеальність» штатних ситуацій функціонування СоПС визначає потребу в альтернативних формалізмах без обмежень сервісу-вузла. Запропоновано їх визначення в руслі підходів до розв'язання циклічних залежностей за допомогою компонента-по-

середника (так званого «проху») в розподілених компонентно орієнтованих застосунках, зокрема відомому каркасі Spring [32].

*Визначення 6.* Опис, на функціональному рівні, семантичного Веб-сервісу-посередника для усунення циклічної залежності  $CD$  – це пара множин його вхідних і вихідних концептів

$$cp = \langle INP; OUP \rangle,$$

які є, відповідно, об'єднаннями підмножин:

а) вхідних концептів сервісів зі складу залежності  $CD$ , семантично зіставлених вихідним концептам компонентних сервісів  $sb$ , які безпосередньо передують  $CD$  в композиції, або вхідним концептам запиту щодо неї ( $INP$ ):

$$INP = \cup \{In^*(sc_i), sc_i \in CD\};$$

$$In^*(sc_i) = \{ic \in In(sc_i) \mid \exists sb \notin CD, \quad (6)$$

$$x \in Out(sb) \cup IN; \quad Exact(x, ic) \quad !$$

$Plugin(x, ic)\};$

б) вихідних концептів сервісів зі складу залежності  $CD$ , семантично зіставлених вхідним концептам компонентних сервісів  $sa$  – безпосередніх наступників залежності  $CD$  у композиції, або вихідним концептам запиту щодо неї ( $OUP$ ):

$$OUP = \cup \{Out^*(sc_i), sc_i \in CD\};$$

$$Out^*(sc_i) = \{ic \in Out(sc_i) \mid \exists sa \notin CD, \quad (7)$$

$$x \in In(sa) \cup Out; \quad Exact(x, ic) \quad !$$

$Subsume(x, ic)\}.$

*Зауваження 1.* Якщо в множині  $OUP$  з (5) жодний концепт не є семантично зіставленим з певним концептом компонентного сервісу поза циклічною залежністю  $CD$  або в (6),(7)  $OUP = \emptyset$ , гілку(и) з  $CD$  слід вилучити з графа композиції.

Згідно з виразами (6),(7), сервіс-посередник долає зазначені обмеження сервіса-вузла за допомогою двох основних відмінностей від нього:

а) відсутності «надлишкової» внутрішньої структури, тобто тих компонентних Веб-сервісів, вихідні концепти яких (з (1)) не є семантично зіставленими з деяким вхідним концептом певного компонентного сервісу поза циклом на гілці без тупиків;

б) урахування в його поданні функціонального рівня лише вхідних концептів тих сервісів у циклі, вхідні концепти яких безпосередньо пов'язані з вихідними кон-

цептами певних сервісів у шарі графа композиції, що передує шару з циклом, або з вхідними концептами запиту щодо композиції, а також вихідних концептів, семантично зіставлених вхідним концептам довільних компонентних сервісів поза циклом, розміщеним на гілках без тупиків.

*Визначення 7.* Спрощення циклічної залежності  $CD$  – її під-граф, отриманий вилученням Веб-сервісів, вихідні концепти яких не є семантично зіставленими з деякими сервісами – наступниками  $CD$ , а також вилученням з множин вхідних і вихідних концептів залишених Веб-сервісів тих концептів, які були семантично зіставлені з вхідними й вихідними концептами вилучених сервісів.

*Зауваження 2.* На відміну від сервісу-вузла, сервіс-посередник і сервіси зі складу спрощення потребують додаткового відшукування або компонування, насамперед базовими методами з їхнього ядра, запропонованого для розвитку моделі АКС [4] на початку статті.

Співвідношення трьох запроваджених формалізмів зручно проілюструвати на де-факто стандартному прикладі композитного сервісу з залежностями для онлайн-магазину [22]. Склад його компонентних сервісів описано в табл. 2 (вхідним концептам у дужках зіставлено сервіси, що їх продукують).

Структуру композитного сервісу показано на рис. 8. Згідно з ним, у композитному сервісі наявні дві циклічні залежності (позначені сірим кольором):

– авто-цикл  $S_3$  ( $S_3$  потребує вихідні концепти від себе й  $S_1$ );

– цикл з п'яти сервісів ( $S_7$ ;  $S_8$ ;  $S_{10}$ ;  $S_{11}$  і  $S_{12}$ ), де  $S_7$  потребує вихідні концепти від  $S_{10}$ ,  $S_{10}$  залежить від  $S_{11}$ , який, у свою чергу, залежить від  $S_8$ , залежного від  $S_{12}$ , тоді як  $S_{12}$  залежить від  $S_7$ .

Результати опрацювання цих залежностей за допомогою сервісу-вузла, сервісу-посередника та спрощення зіставлені на рис. 9. Ліворуч показано сервіси-вузли  $CN_1$ ,  $CN_2$  і сервіси-посередники  $SP1, SP2$  (для авто-циклу вузол  $CN1$  і посередник  $SP1$  збігаються), а праворуч – чотири сервіси  $SS3, SS7, SS8, SS10$  зі складу спрощення циклу ( $S_7$ ;  $S_8$ ;  $S_{10}$ ;  $S_{11}$ ;  $S_{12}$ ), отримані вилученням з

Таблиця 2. Веб-сервіси для ілюстративного сценарію онлайн-магазину

№	Веб-сервіс	Входи	Виходи
$S_1$	Онлайн каталог	Вибір елемента (запит користувача)	Деталі={Вибір категорії, код товару, Вибір кількості, Ціна товару, Вартість, IP адреса}
$S_2$	Розміщення клієнта	IP адреса ( $S_1$ )	Розміщення клієнта
$S_3$	Перевірка позицій	Вибір (категорії, код товару, кількість) ( $S_1$ ), Номер товару ( $S_3$ )	Наявність, кількість
$S_4$	Податки	Вибір категорії, вибір ціни ( $S_1$ )	Сума податку
$S_5$	Страховання	Вибір категорії, вибір ціни ( $S_1$ ), Вибір кількості, Вартість ( $S_1$ )	Довідка про страхування, вартість
$S_6$	Отримання	Наявність ( $S_3$ )	Розташування
$S_7$	Розрахунок ціни	Ціна товару ( $S_1$ ), Сума податку ( $S_4$ ), Вартість страхування ( $S_4$ ), Вартість доставки ( $S_{10}$ )	Загальна ціна
$S_8$	Формування замовлення	Узгодження деталей ( $S_1$ ), Загальна ціна (грн.) ( $S_{12}$ )	Замовлення на придбання
$S_9$	Прогноз погоди	Розміщення предмету ( $S_6$ ), Місцезнаходження клієнта ( $S_2$ )	Погода
$S_{10}$	Доставка	Погода( $S_9$ ), Узгодження деталей ( $S_1$ ), Квитанція ( $S_{11}$ ),	Код доставки, вартість, номер відстеження
$S_{11}$	Оплата	Замовлення на придбання ( $S_8$ )	Квитанція
$S_{12}$	Конвертер валют	Загальна ціна ( $S_7$ )	Загальна ціна в національній валюті

їх множин вхідних і вихідних концептів тих концептів, які були семантично зіставлені з вхідними й вихідними концептами «непродуктивних» вилучених сервісів  $S_{11}$ ,  $S_{12}$ . Хоча структури композитного сервісу-вузла та сервісу-посередника однакові, множини їх вхідних і вихідних концептів відмінні між собою в сенсі визначень 5, 6.

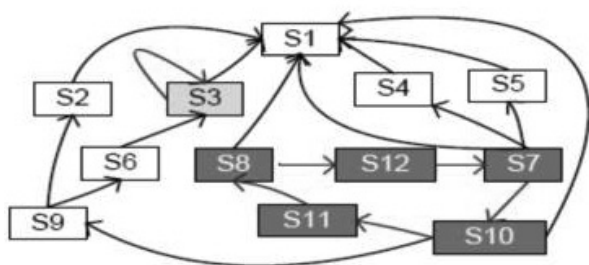


Рис. 8. Два цикли в композитному сервісі

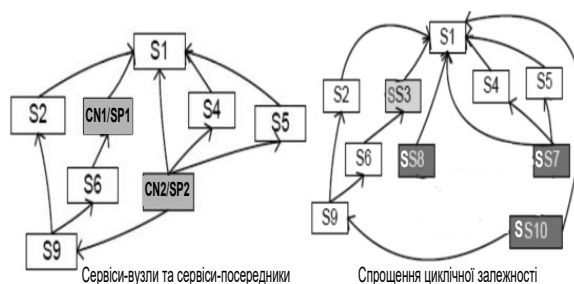


Рис. 9. Опрацювання циклічної залежності: вузол, посередник, спрощення

Як показано на рис. 9, сервіс-вузол замінює циклічну залежність, інкапсулюючи її внутрішню структуру; сервіс-посередник «вирізує» цю структуру без порушення цілісності; спрощення «розриває» залежність, залишаючи лише продуктивні компонентні сервіси з їхніми функціонально необхідними зв'язками.

### Висновки

Запропоновано стале забезпечення якості адаптивного композитного семантичного Веб-сервісу за допомогою OWL-S-специфікації контексту виконання його самого і компонентних Веб-сервісів та узгодженого застосування специфікованого контексту на всіх етапах компонування, а також формалізмів сервісу-посередника й спрощення для усунення циклічних залежностей у композиції в ситуаціях неефективності попередньо запропонованого формалізму сервісу-вузла. Обґрунтовано застосування спеціального процесного числення контекстно-залежних амбієнтів (Ф.Сів) для засвідчення якості формованого адаптивного композитного семантичного Веб-сервісу за допомогою його динамічної верифікації, насамперед перевірки властивостей відповідності контексту запитувача та живості – «композитний сервіс завжди надає запитаний користувачем результат».

Застосування розроблених формалізмів сприяє підвищенню рівня відповідності композиції очікуванням запитувачів та уможливорює багаторазове уточнення базової композиції функціонального рівня, формованої на початку конструювання адаптивного композитного Веб-сервісу, на підтримку (не) передбачено змінних розподілених ділових процесів сучасних організацій.

### Література

1. Андон П.І. Збіркове програмування компонентних і сервіс-орієнтованих прикладних програмних систем / П.І. Андон, О.О. Слабоспицька // Проблеми програмування. – 2017. – № 3 – С. 31–51.
2. Weyns D. et al. Perpetual Assurances for Self-Adaptive Systems. [Electronic resource]. Mode of access: <https://arxiv.org/ftp/arxiv/papers/1903/1903.04771.pdf>.
3. De Lemos R. Software engineering for self-adaptive systems: Research challenges in the provision of assurances / R De Lemos, D Garlan, C Ghezzi et al. (Eds.) // Self-Adaptive Systems III LNCS 9640. – 2017. – P. 3-30.
4. Слабоспицька О.О. Рамкова модель адаптивного композитного сервісу в семантичному Веб-середовищі / О.О.Слабоспицька // Проблеми програмування. – 2017. – №4. – С. 51-65.
5. Слабоспицька О.О. Уніфікований процес композиції адаптивного сервісу в семантичному Веб-середовищі / О.О.Слабоспицька // Проблеми програмування. – 2018. – № 1. – С. 65–76.
6. Rodriguez Mier P. An Integrated Semantic Web Service Discovery and Composition Framework / P Rodriguez Mier, C Pedrinaci, M Lama et al. //IEEE Trans. on Services Comp. – 2015. – V.9. – Is.4. – P. 537–555.
7. Bansal S. Generalized semantic Web service composition / S.Bansal, A.Bansal, G.Gupta, M.Brian Blake// Service Oriented Computing and Applications. – 2016. – V.10. – Is. 2. – P. 111–133.
8. Ben Lamine R. A Framework for the composition and formal verification of adaptable semantic Web services / R.Ben Lamine, R.Ben Jemaa, I.Amous // Proc. of ACM MoMM conf. (MoMM'18). – ACM, Yogyakarta, Indonesia, 2018. – 9 p.
9. Ben Lamine R. Formal Specification of Adaptable Semantic Web Services Composition // Int. J. of Information Technology and Web Engineering. 2018. – N 13(4). – P. 14–34.
10. Ben Lamine R. Graph Planning Based Composition For Adaptable Semantic Web Services / R.Ben Lamine, R.Ben Jemaa, I.Amous // Proc. of the 2017 Int. Conf. on Knowledge Based and Intelligent Information and Engineering Systems, KES2017. – 2017, Marseille, France. – P. 358–368.
11. Alférez G.H. Achieving autonomic Web service compositions with models at runtime / G.H.Alférez, V.Pelechano / Computers & Electrical Engineering. – 2017. – V. 63. – P/ 332-352.
12. Pistore M. A Minimalist Approach to Semantic Annotations for Web Processes Compositions / M.Pistore, L.Spalazzi, P.Traverso // Proc. European Semantic Web Conf. ESWC 2006. – 2006. – P. 620-634.
13. Bucchiarone A. et al. Domain Objects for Dynamic and Incremental Service Composition / A.Bucchiarone, M. De Sanctis, M. Pistore // In: Villari M.et al. (Eds.) Proc. ESOC 2014, Manchester, UK – LNCS 8745, 2014. – P. 62–80.
14. Yu L. Goal-driven context-aware service composition / L.Yu, A.Glenstrup, Y.Zhang et al. – Pervasive Computing and Applications, ICP-CA 2010. – 2010. – P. 342–347.
15. Li L. Semantic based aspect-oriented programming for context-aware Web service composi-

- tion / L.Li, D.Liu, A.Bouguettaya // Information Systems – 2011. – V. 36(3). – P. 551–564.
16. Baidouri H. Towards a Context-Aware Composition of Services / H.Baidouri, H.Hafiddi, M.Nassar et al. // (2012). . Int.J. of Computer Science and Network Security. – 2012. – V. 12(3). – P. 133–140.
  17. Mcheick H. Modele de composition des services Web semantiques par orchestration a la demande / H.Mcheick // Proc. of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE. 2012), Montreal, Canada. [Electronic resource]. Mode of access: [https://www.researchgate.net/publication/257527267\\_Modele\\_de\\_Composition\\_Des\\_Services\\_Web\\_Semantiques\\_par\\_Orchestration\\_a\\_la\\_demande](https://www.researchgate.net/publication/257527267_Modele_de_Composition_Des_Services_Web_Semantiques_par_Orchestration_a_la_demande).
  18. Furno A. Context-aware composition of semantic web services / A.Furno, E.Zimeo // Mobile Networks and Applications. – 2014. – V. 19(2) – P. 235–248.
  19. Yan Y. Anytime QoS-aware service composition over the GraphPlan / Y. Yan, M.Chen // Service Oriented Computing and Applications. – 2015. – V. 9(1). –, P. 1–19.
  20. Boukadi K. Specification and Verification of Views over Composite Web Services Using High Level Petri-Nets / K.Boukadi, C.Ghedira, Z.Maamar et al. // Proc. of the 9th International Conference on Enterprise Information System, ICEIS 2007. – P. 107–112.
  21. Madkour M. Context-Aware Service Adaptation: An Approach Based on Fuzzy Sets and Service Composition / M.Madkour, I.El Ghanami, A.Maach // J. of Inform. Sci. and Eng. – 2013. – V. 29. – P. 116–126.
  22. Omer A.M. Automatic management of cyclic dependency among web services / A.M.Omer, A.Schill // Proc. of IEEE Int. Conf. on Computational Science and Engineering. – 2011. – P. 44–51.
  23. Siewe F. The Calculus of Context-aware Ambients (CCA): How to program: Capabilities and Processes / F.Siewe – 2017. [Electronic resource]. Mode of access: [http://www.cse.dmu.ac.uk/~fsiewe/ccaPL/ccaPL\\_processes.pdf](http://www.cse.dmu.ac.uk/~fsiewe/ccaPL/ccaPL_processes.pdf).
  24. Сторінка «Calculus of Context-aware Ambients (CCA)» Дослідницької лабораторії технології програмного забезпечення Університету ДеМонфору (Лейчестер, Велика Британія). Електронний ресурс. Режим доступу: <http://www.cse.dmu.ac.uk/~fsiewe/fscca.html>.
  25. Najar S. Service Discovery Mechanism for an Intentional Pervasive Information System / S.Najar, M.Kirsch-Pinheiro, C.Souveyet et al. // IEEE 19th Int. Conference on Web Services (ICWS). – 2012. – P. 520–527.
  26. Cherif S. An Integrated context-aware Planning Approach to Self-Adaptation Web Service Composition / S.Charif, R.Ben Djemaa, I.Amous // Proc. of the 14th Int. Conf. on Advances in Mobile Computing and Multi Media (MoMM '16), 2016. – P. 3–11.
  27. Martin D. Bringing semantics to web services: The OWL-S approach / D. Martin, M.Paolucci et al. // Proc First Int Work Semant Web Serv Web Process Compos SWSWPC 2004. – LNCS, V. 3387, 2005. – P. 26-42.
  28. Kirsch-Pinheiro M. (2008). Context-aware service selection using graph matching / M. Kirsch-Pinheiro // 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop. – 2008. [Electronic resource]. Mode of access: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-411/paper7.pdf>.
  29. Cherif S. Adaptable Web Service Registry for Publishing Context Aware Service Composition / S.Charif, R.Ben Djemaa, I.Amous. I. // Proc. of the 17th Int. Conf. on Information Integration and Web-based Applications & Services (IIWAS'15), Brussels, Belgium. – 2015. [Electronic resource]. Mode of access: <https://dl.acm.org/doi/pdf/10.1145/2837185.2837189?download=true>.
  30. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем / Ю.Г.Карпов – СПб.:БХВ-Петербург, 2010. – 560 с.
  31. Siewe F. ccaPL: a Programming Language for the Calculus of Context-aware Ambients / F.Siewe – 2012. [Electronic resource]. Mode of access: [citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.7085&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.7085&rep=rep1&type=pdf).
  32. Динеш Р. Spring. Все паттерны проектирования / Р.Динеш – Издательский дом «Питер», 2018. – 320 с.

## References

1. Andon P.I. Assembly programming component and service-oriented applied software systems / P.I Andon, O.O. Slabospitska // Problems in Programming. – 2017. – № 3 – P. 31–51.
2. Weyns D. et al. Perpetual Assurances for Self-Adaptive Systems. [Electronic resource].

- Mode of access: <https://arxiv.org/ftp/arxiv/papers/1903/1903.04771.pdf>.
3. De Lemos R. Software engineering for self-adaptive systems: Research challenges in the provision of assurances / R De Lemos, D Garland, C Ghezzi et al. (Eds.) // *Self-Adaptive Systems III LNCS 9640*. – 2017. – P. 3-30.
  4. Slabospitska O.O. Reference model for Semantic Web adaptive composite service / O.O Slabospitska // *Problems in Programming*. – 2017. – № 4 – P. 51–65.
  5. Slabospitska O.O. Unified process for adaptive Semantic Web service composition / O.O Slabospitska // *Problems in Programming*. – 2018. – № 1. – P. 65–76.
  6. Rodriguez Mier P. An Integrated Semantic Web Service Discovery and Composition Framework / P Rodriguez Mier, C Pedrinaci, M Lama et al. // *IEEE Trans. on Services Comp.* – 2015. – V.9. – Is.4. – P. 537–555.
  7. Bansal S. Generalized semantic Web service composition / S.Bansal, A.Bansal, G.Gupta, M.Brian Blake // *Service Oriented Computing and Applications*. – 2016. – V.10. – Is. 2. – P. 111–133.
  8. Ben Lamine R. A Framework for the composition and formal verification of adaptable semantic Web services / R.Ben Lamine, R.Ben Jemaa, I.Amous // *Proc. of ACM MoMM conf. (MoMM'18)*. – ACM, Yogyakarta, Indonesia, 2018. – 9 p.
  9. Ben Lamine R. Formal Specification of Adaptable Semantic Web Services Composition // *Int. J. of Information Technology and Web Engineering*. 2018. – N 13(4). – P. 14–34.
  10. Ben Lamine R. Graph Planning Based Composition For Adaptable Semantic Web Services / R.Ben Lamine, R.Ben Jemaa, I.Amous // *Proc. of the 2017 Int. Conf. on Knowledge Based and Intelligent Information and Engineering Systems, KES2017*. – 2017, Marseille, France. – P. 358–368.
  11. Alférez G.H. Achieving autonomic Web service compositions with models at runtime / G.H.Alférez, V.Pelechano / *Computers & Electrical Engineering*. – 2017. – V. 63. – P/ 332-352.
  12. Pistore M. A Minimalist Approach to Semantic Annotations for Web Processes Compositions / M.Pistore, L.Spalazzi, P.Traverso // *Proc. European Semantic Web Conf. ESWC 2006*. – 2006. – P. 620-634.
  13. Bucchiarone A. et al. Domain Objects for Dynamic and Incremental Service Composition / A.Bucchiarone, M. De Sanctis, M. Pistore // In: Villari M.et al. (Eds.) *Proc. ESOC 2014*, Manchester, UK – LNCS 8745, 2014. – P. 62–80.
  14. Yu L. Goal-driven context-aware service composition / L.Yu, A.Glenstrup, Y.Zhang et al. – *Pervasive Computing and Applications, ICP-CA 2010*. – 2010. – P. 342–347.
  15. Li L. Semantic based aspect-oriented programming for context-aware Web service composition / L.Li, D.Liu, A.Bouguettaya // *Information Systems* – 2011. – V. 36(3). – P. 551–564.
  16. Baidouri H. Towards a Context-Aware Composition of Services / H.Baidouri, H.Hafiddi, M.Nassar et al. // (2012). . *Int.J. of Computer Science and Network Security*. – 2012. – V. 12(3). – P. 133–140.
  17. Mcheick H. Modele de composition des services Web semantiques par orchestration a la demande / H.Mcheick // *Proc. of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE. 2012)*, Montreal, Canada. [Electronic resource]. Mode of access: [https://www.researchgate.net/publication/257527267\\_Modele\\_de\\_Composition\\_Des\\_Services\\_Web\\_Semantiques\\_par\\_Orchestration\\_a\\_la\\_demande](https://www.researchgate.net/publication/257527267_Modele_de_Composition_Des_Services_Web_Semantiques_par_Orchestration_a_la_demande).
  18. Furno A. Context-aware composition of semantic web services / A.Furno, E.Zimeo // *Mobile Networks and Applications*. – 2014. – V. 19(2) – P. 235–248.
  19. Yan Y. Anytime QoS-aware service composition over the GraphPlan / Y. Yan, M.Chen // *Service Oriented Computing and Applications*. – 2015. – V. 9(1). –, P. 1–19.
  20. Boukadi K. Specification and Verification of Views over Composite Web Services Using High Level Petri-Nets / K.Boukadi, C.Ghedira, Z.Maamar et al. // *Proc. of the 9th International Conference on Enterprise Information System, ICEIS 2007*. – P. 107–112.
  21. Madkour M. Context-Aware Service Adaptation: An Approach Based on Fuzzy Sets and Service Composition / M.Madkour, I.El Ghanami, A.Maach // *J. of Inform. Sci. and Eng.* – 2013. – V. 29. – P. 116–126.
  22. Omer A.M. Automatic management of cyclic dependency among web services / A.M.Omer, A.Schill // *Proc. of IEEE Int. Conf. on Computational Science and Engineering*. – 2011. – P. 44–51.

23. Siewe F. The Calculus of Context-aware Ambients (CCA): How to program: Capabilities and Processes / F.Siewe – 2017. [Electronic resource]. Mode of access: [http://www.cse.dmu.ac.uk/~fsiewe/ccaPL/ccaPL\\_processes.pdf](http://www.cse.dmu.ac.uk/~fsiewe/ccaPL/ccaPL_processes.pdf).
24. Сторінка «Calculus of Context-aware Ambients (CCA)» Дослідницької лабораторії технології програмного забезпечення Університету ДеМонфору (Лейчестер, Велика Британія). Електронний ресурс. Режим доступу: <http://www.cse.dmu.ac.uk/~fsiewe/fscsa.html>.
25. Najar S. Service Discovery Mechanism for an Intentional Pervasive Information System / S.Najar, M.Kirsch-Pinheiro, C.Souveyet et al. // IEEE 19th Int. Conference on Web Services (ICWS). – 2012. – P. 520–527.
26. Cherif S. An Integrated context-aware Planning Approach to Self-Adaptation Web Service Composition / S.Cherif, R.Ben Djemaa, I.Amous // Proc. of the 14th Int. Conf. on Advances in Mobile Computing and Multi Media (MoMM '16), 2016. – P. 3–11.
27. Martin D. Bringing semantics to web services: The OWL-S approach / D. Martin, M.Paolucci et al. // Proc First Int Work Semant Web Serv Web Process Compos SWSWPC 2004. – LNCS, V. 3387, 2005. – P. 26-42.
28. Kirsch-Pinheiro M. (2008). Context-aware service selection using graph matching / M. Kirsch-Pinheiro // 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop. – 2008. [Electronic resource]. Mode of access: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-411/paper7.pdf>.
29. Cherif S. Adaptable Web Service Registry for Publishing Context Aware Service Composition / S.Cherif, R.Ben Djemaa, I.Amous. I. // Proc. of the 17th Int. Conf. on Information Integration and Web-based Applications & Services (IIWAS'15), Brussels, Belgium. – 2015. [Electronic resource]. Mode of access: <https://dl.acm.org/doi/pdf/10.1145/2837185.2837189?download=true>.
30. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем / Ю.Г.Карпов – СПб.:БХВ-Петербург, 2010. – 560 с.
31. Siewe F. ccaPL: a Programming Language for the Calculus of Context-aware Ambients / F.Siewe – 2012. [Electronic resource]. Mode of access: [citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.7085&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.725.7085&rep=rep1&type=pdf).
32. Dinesh R. Spring 5 Design Patterns. / R. Dinesh – Publishing house «Piter», 2018. – 320 p.

Отримано 28.11.2021

### *Про авторів:*

Андон Пилип Іларіонович,  
академік НАН України, д-р ф.-м. н.,  
директор,  
кількість публікацій в українських  
виданнях – більше 400,  
кількість публікацій у закордонних  
виданнях – 10,  
номер ORCID – 0000-0002-2204-1554

Слабоспицька Ольга Олександрівна,  
кандидат фізико-математичних наук,  
с.н.с., ст. наук. співроб.,  
кількість публікацій в українських  
виданнях – більше 50,  
кількість публікацій у закордонних  
виданнях – 7,  
номер ORCID – 0000-0001-6556-0947

### *Місце роботи авторів:*

Інститут програмних систем  
НАН України,  
03187, Київ-187,  
Проспект Академіка Глушкова, 40.  
Тел.: +38(044) 526 4286.  
E-mail: [olsips2017@gmail.com](mailto:olsips2017@gmail.com)