УДК 004.94

ВИКОРИСТАННЯ ТЕХНІК AI-ПЛАНУВАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧ КОМПОЗИЦІЇ ВЕБ-СЕРВІСІВ

О.В. Захарова

Автоматична композиція сервісів, що представлені процесною моделлю, складна, але дуже актуальна задача, вирішення якої потребує, перш за все строгої формалізації та суттєвої семантизації опису сервісу. Схожість визначення задач інтелектуального планування до задач композиції сервісів доводить можливість застосування підходів AI-планування до вирішення проблем композиції Веб-сервісів, за умови вирішення низки проблем. В даній роботі проводиться аналіз схожості та розбіжностей задач HTN- планування та композиції Веб-сервісів, що представлені процесною моделлю, а саме BPEL- сервісів, визначаються основні проблеми щодо безпосереднього застосування технік AI-планування та пропонуються підходи до їх вирішення, шляхом інтеграції дескриптивної логіки та методів HTN-планування, а також пропонуються підходи до трансляції BPEL описів у HTN-DL.
Ключові слова: семантичний Веб-сервіс, дескриптивна логіка, процесна модель, композиція семантичних Веб-сервісів, задача AI-планування, HTN-планування, BPEL-процеси, HTN-DL.

Автоматизированная композиция сервисов, представленных процессной моделью, является сложной, но на сегодняшний день очень насущной задачей. Ее решение требует, прежде всего, строгой формализации и сильной семантизации представления сервиса. Подобие определений задач интеллектуального планирования и задач композиции сервисов доказывает возможность применения подходов AI-планирования к решению проблем композиции Веб-сервисов, при условии решения ряда имеющихся проблем. В данной работе проводится анализ схожести и различий задач HTN- планирования и композиции Веб-сервисов, представленных процессной моделью, а именно BPEL- сервисов, определяются основные проблемы, возникающие при непосредственном применении техник AI-планирования и предлагаются подходы к их решению путем интеграции дескриптивной логики и методов HTN-планирования, а также предлагаются подходы к трансляции BPEL описаний в HTN-DL.
Ключевые слова: семантический Веб-сервис, дескриптивная логика, модель процесса, композиция семантических Веб-сервисов, задача AI-планирования, HTN-планирование, BPEL-процессы, HTN-DL.

Automated composition of services described by their process model is difficult but very vital task. Its decision needs strict formalization and strong semantization of a service. Similarity of definitions of intelligent planning tasks and services composition objectives demonstrates the possibility of using AI-planning approaches for resolving Web services composition problems, provided a number of solutions to existing problems. In this article the analysis of task similarity and differences of HTN-planning and composition of Web-services presented by process model is executed. BPEL-services are considered. It is defined main problems that appears when AI-planning methods are used and it is proposed approaches for their resolving by integration DL and HTN-planning. Translation algorithms from BPEL to HTN-DL is also discussed.
Key words: semantic Web-service, descriptive logic, process model, semantic Web-service composition, AI-planning task, HTN-planning, BPEL-processes, HTN-DL.

Вступ

Композиція Веб-сервісів має дві цілі: з одного боку – задовільнити складні вимоги клієнта, а з іншого - зменшити складність розробки веб-сервісу, щоб задовільнити складну прикладну систему.

Композиція – це процес, що дозволяє сервісам взаємодіяти інтелігентним шляхом, щоб виявити інші сервіси, забезпечити спілкування між ними та композувати їх у більш складні процеси. Різні запропоновані визначення сходяться в тому, що композиція Веб-сервісів – це можливість забезпечити нову функціональність, отриману з комбінації різних Веб-сервісів, що надаються різними провайдерами. Веб-сервіси можна класифікувати щодо моделей їх взаємодії:

– атомні веб-сервіси, що розглядаються як «чорна скринька», тобто описуються своїми параметрами (вхід, вихід, передумови, після умови);

– поведінкові, Веб-сервіси, що базуються на поведінковій (процесній) моделі, часто називаються «сірою скринькою», описуються порядком виконання операцій.

Можна виділити два типи процесних моделей:

– сервіси, що базуються на виконанні – в цій моделі поведінка веб-сервісу описується лінійною послідовністю дій, кожна дія описує одне виконання, але в кожній послідовності кожна точка виконання має лише одного послідовника, та

– сервіси, що засновані на деревах, вони ідентичні структурі зв'язків, яка описує поведінку можливого виконання, де кожна точка виконання може мати декілька послідовників.

Існують синтаксичні та семантичні підходи до вирішення задачі композиції. До прикладів семантичних підходів та технік до вирішення задачі композиції можна віднести техніки, що базуються на AI-плануванні [1], підходи, що базуються на алгоритмах ланцюжка, техніки композиції на основі інтерфейсу автоматів. Ефективність вирішення задач Веб-сервісів досягається їх семантизацією. Щоб семантично описати Веб-сервіс, використовуються анотації. Сервіс анотується семантичною моделлю, яка забезпечує

семантичну інформацію щодо його функціональності (входи, виходи). Процес анотування полягає у виділенні функціональних компонент та встановлення зв'язку кожної функціонального компонента з відповідним елементом семантичної моделі. Для представлення Веб-сервісу може бути використана будь-яка мова специфікації Веб-сервісів.

Теоретичні основи інтелектуального планування

Під задачею AI (інтелектуального) планування [2, 3] розуміють процес пошуку плану досягнення цілі. Щоб надати формальне визначення задачі з точки зору штучного інтелекту, необхідно ввести базові поняття, що мають безпосереднє відношення до процесу планування.

Планування – це пошук послідовності дій, що ведуть до визначеної мети. Цілеспрямована послідовність дій не може розглядатися окремо від контексту їх виконання. Послідовність дій призведе або не призведе до поставленої цілі залежно від зовнішніх умов. Тобто виконання будь-якого плану оточене контекстом, в якому цей план виконується. Модель контексту, в якому виконуються дії, називається *моделлю світу* (world model).

Модель світу може включати статичну та динамічну складові. До статичних елементів відносять моделі об'єктів світу та відношень між ними. Динамічним елементом є модель змінення світу за своїми внутрішніми законами, не залежно від впливу з боку суб'єкта, що виконує план. Динамічна складова називається *динамікою світу* (world dynamics).

Дії плану та динаміка світу здійснюють перетворення моделі світу (як правило, лише статичної складової). Про змінення моделі світу кажуть, що модель світу переходить у новий *стан* (state). Стан моделі світу у момент, що безпосередньо передує моменту початку виконання плану, називається *початковим станом* (initial state) задачі планування. Більшість сучасних планувальників при побудові плану оперують саме поняттям стану. Формально, стан може бути представлений різними способами.

Конструктивним елементом для процесу планування є дія (модель дії.) *Дія* (action) визначає, які зміни відбудуться в моделі світу, якщо дія буде виконана. Як модель дії в інтелектуальному плануванні використовується сутність, яка описує вид діяльності, умови, коли ця дія може бути виконана, та ефекти, які ця дія виробляє. Дію прийнято позначати у вигляді кортежу, що складається з трьох елементів: <*Name, Precondition, Effect*>, де *Name* – це ім'я дії, тобто символьне позначення деякого виду діяльності, *Precondition* – передумова дії – ті умови, при яких можливе виконання дії, *Effect* – ефект дії – опис змін, які виробляє дія.

Задача планування полягає у пошуку послідовності дій, застосування якої у початковому стані моделі світу призведе до такого стану, в якому досягається завчасно задана ціль.

Послідовність дій, що отримана в результаті вирішення задачі планування, називається планом (plan). Програмний засіб, що здійснює рішення задачі планування, називається планувальником (planner). Важливо зазначити, що планувальник лише шукає план, але не відповідає за його виконання.

Окрім наведених вище понять є ще одне важливе поняття, яке часто зустрічається в описах задач планування. Це – домен планування (planning domain). Це поняття складно визначити формально, але у де-якому сенсі це урізаний опис предметної області, в межах якого здійснюється планування. Домен планування складається з описів видів дій (які є дозволеними в даній моделі), описів динаміки та законів світу, а також описів можливих видів відношень між об'єктами. Але опис самих об'єктів моделі світу не входить до домена планування. Тому його неможна вважати синонімом "предметної області".

Залежно від умов, в яких відбувається процес планування, розрізняють декілька різновидів середовищ планування (planning environment). Вони визначаються низкою властивостей виконавця та світу, в якому він оперує, та формально представляються кортежем з п'яти елементів: <*World observability, World static character, Action determinancy, Action durability, Action parallelism*>, де *World observability* – спостережуваність світу (повністю або не повністю), *World static character* – статичність світу (статичний або динамічний), *Action determinancy* – детермінованість дій (детерміновані або ні), *Action durability* — тривалість дій (тривалі або дискретні), *Action parallelism* – паралелізм дій (є допустимим чи ні).

Різні способи надання значень елементам цього кортежа породжує різні середовища планування. Класичним плануванням вважають планування в такому середовищі, коли світ повністю спостерігається та є статичним, а дії детерміновані та дискретні, паралельне виконання дій не дозволяється.

HTN-планування (Hierarchical Task Network) подібне до класичного планування в тому сенсі, що кожний стан світу представляється як множина атомів, і кожна дія відповідає детерміністичному переходу зі стану в стан. Однак, HTN планувальники відрізняються від класичних планувальників тим, що вони планують та для чого вони планують. Метою такого планувальника є не досягнення множини цілей, а виконання де-якої множини задач. Вхід до системи планування включає множину операторів та множину методів, кожен з яких є приписанням, як декомпозувати де-яку задачу в множину підзадач (дрібніших задач). Планування виконується шляхом декомпозиції непримітивних задач рекурсивно на дрібніші та й дрібніші підзадачі, доки не досягаються примітивні задачі, які можуть бути виконані безпосередньо, використовуючи оператори планування. HTN планувальник буде застосовувати оператор лише якщо це передбачено методом. Таким чином, кожний метод є частиною специфікації Branch функції.

Доцільність застосування методів AI планування для вирішення задач Веб-сервісів

Наведений вище стислий огляд задач AI планування демонструє їх схожість з проблемою композиції Веб-сервісів. Не складно примітити й відповідність основних понять. Так, атомний сервіс або атомний елемент складного сервісу, що представлений процесною моделлю, подібний до оператора планування в системі понять AI-планування, а його виконання переводить систему зі стану в стан. *Ціль* – це формула, що описує гіпотетичний сервіс, який необхідно побудувати для вирішення конкретної бізнес-задачі. Сервіси, процедурна поведінка яких задана за допомогою BPEL, легко трансформуються в систему перехідних станів (STS). А більшість підходів планування покладаються на загальну модель, модель системи перехідних станів. У системі перехідних станів існує кінцева або рекурсивно перелічувальна множина станів, дій та подій разом з функцією переходів, яка відображує трійку (стан, дія, подія) у множину станів. Якщо задана функція переходів, то мета планування – знайти, яку дію потрібно застосувати до яких станів, щоб досягти де-яких цілей, виходячи з деякої заданої ситуації. Аналогічно, при вирішенні задачі композиції Веб-сервісів нам треба визначити, який сервіс потрібно застосувати та до якого стану, щоб побудувати результуючий сервіс. Тобто, ми можемо Веб-сервіси транслювати у оператори планування, ціль виразити логічною умовою, і тоді використати планувальник, щоб згенерувати план, який по суті є послідовністю екземплярів Веб-сервісів, а саме, послідовною композицією, що призводить до того, щоб цільова умова була вірною при їх виконанні.

Але, таке пряме кодування композиції Веб-сервісів як задачі планування має декілька проблем. По-перше, звичайна логіка, що використовуються для представлення передумов та ефектів в системі планування має радикально іншу виразну потужність ніж мови Веб-онтологій. В системі планування, стан представлення світу містить лише факти про світ, але не містить аксіом. Але аксіоми є суттєвими в задачах композиції Веб-сервісів, щоб допомогти системі встановити відношення між складними концептами, що надасть можливість зв'язати поняття з різних онтологій.

Ще одне питання – це Твердження Замкненого Світу, на яке спираються системи AI планування. Якщо факт не міститься в локальній базі знань, то він просто вважається не вірним. Однак, це не має місця в Веб-контексті, так як ми не можемо очікувати, що матимемо всю релевантну інформацію в нашій локальній базі знань. При побудові рішення ми повинні розглядати відомі факти та можливості як сумісні з цими фактами. Щоб вирішити задачу композиції, планувальник, який володіє не повною інформацією, повинен збирати де-яку інформацію. Необхідно, щоб збір інформації та генерація композиції взаємодіяли таким чином, щоб можна було прийняти рішення стосовно того, який сервіс включити до композиції. Враховуючи розміри та природу Веб, намагання зібрати всю можливу інформацію є, в кращому випадку, марнотратством, а в загальному – практично не можливим. Релевантність можливої інформації повинна визначатися контекстом задачі композиції та можливих комбінацій, які розглядає планувальник в тому сенсі, чи доцільно збирати інформацію в цій точці.

Також необхідно, щоб система планування відрізняла сервіси, які надають інформацію, від сервісів, що роблять певні зміни в світі. Необхідно описати, чи сервіс при виконанні просто забезпечує інформацію, чи має інші ефекти, які впливають на світ. Інформаційні сервіси планувальник може виконувати вільно, так як вони не змінюють нічого окрім наших знань. Але виконання іншого типу сервісів потребує більше уваги, оскільки це може принести користувачеві не бажаний результат. Слід зазначити, що такі сервіси також можуть надавати де-яку інформацію в якості результату виконання. Дійсно, будь-який сервіс, що має вихідні параметри або ефекти, забезпечує інформацію. Більш того, один й той самий сервіс може мати і ефекти змінення світу і інформаційні ефекти, які повинні бути ідентифіковані.

Представлення Веб-сервісу як оператора планування має ще дві проблеми:

1) неможливо описати внутрішню структуру композитного сервісу;

2) оператор планування має лише інформацію про передумови та ефекти, але нам необхідно мати можливість виражати інформацію про самі сервіси, таку як провайдер сервісу, повноваження провайдера, оцінки сервісу користувачами та інші.

Слід зазначити, що для автоматизації задачі композиції Веб-сервісів було запропоновано навіть декілька технік AI планування. Але більшість з цих систем базуються на казуальному плануванні, де розглядаються лише примітивні дії, та нема складних дій. Представлення стану також обмежене таким чином, щоб встановити заземлені атоми та не розглядати аксіоми домену.

HTN-DL

Для запобігання описаних вище проблем в [4] був представлений формалізм планування HTN-DL, що поєднує формалізм планування HTN з представленням дескриптивною логікою (ДЛ). Слід зазначити, що HTN-DL – це не перша спроба поєднати дескриптивні логіки та алгоритми планування. Існує декілька різних підходів до комбінування ДЛ та систем планування. ДЛ, перш за все, використовувалися для представлення станів світу, а дії використовувалися в плануванні, і генерувалися плани. В даному випадку, найбільш загальним підходом використання ДЛ для представлення стану світу є пропозиційне представлення задач планування, де опис ДЛ концептів представляє різні стани (наприклад CLASP системи [5]). Однак, ці підходи

базувалися на твердженні замкненого світу та потребували повноти знань про світ [6]. Щоб якось обійти це обмеження, можна було б використовувати ДЛ, яка об'єднує не монотонні оператори, а саме, оператор мінімальних знань та оператор припущення за замовченням [7, 8]. Але не існує резонерів, які підтримують таку виразність.

HTN-DL пропонує використовувати дескриптивну логіку як виразну мову представлення знань для опису як станів так і дій. Ієрархічна структура доменів HTN планування дозволяє зручно визначати описи композитних сервісів. Композитні Веб-сервіси можуть бути відображені в HTN методи, а атомні Веб-сервіси – у HTN оператори. Домени HTN-стилю добре вписуються у слабопов'язану природу Веб-сервісів: різні декомпозиції задачі є незалежними, тому проектувальник методу не повинен мати точних знань про те, як буде відбуватися подальша декомпозиція та як відбувалась попередня. Однак, HTN планування не вирішує більшості проблем, що були описані вище. Більше того, існують інші обмеження, специфічні для HTN планування, що робить складним його застосування до вирішення проблеми композиції Веб-сервісів:

– у HTN плануванні задача ідентифікується лише ім'ям та кількістю аргументів. Цього не достатньо, щоб виразити або вивести відношення між різними задачами;

– існує чітке розмежування між примітивними та не примітивними задачами, що не може бути застосовано до Веб-сервісів;

– існує відображення між операторами та примітивними задачами один-до-одного, тобто існує лише один оператор, який виконується для будь-якої примітивної задачі. Зрозуміло, що це для Веб-сервісів надто сильне обмеження.

Щоб подолати ці проблеми запропонований формалізм HTN-DL комбінує HTN планування з ДЛ онтологіями. Ключові відмінності HTN-DL щодо систем класичного HTN планування можна класифікувати у декілька наступних категорій:

– описи задач. Задачі описуються за допомогою онтологій та відповідають задачам з операторами та методами, що зроблені на основі онтології задач. Символи задач представляються ДЛ-концептами, а оператори та методи представляються екземплярами. Відповідність задач частково скорочується до задачі пошуку екземплярів в ДЛ. Окрім цього, передумови та ефекти пов'язані із задачами, таким чином забезпечується більше інформації, яка також використовується для визначення відповідності сервісів;

– описи методів/операторів. Визначення оператора та метода використовують ДЛ –запити для опису умов у передумовах та ефектах. Виходи сервісів, які традиційно не розглядаються у системах планування, представляються як екзистенціальні змінні в описах ефектів. Ефекти знань дії виражаються окремо, так, що можна розрізнити інформаційні та сервіси, що змінюють світ;

– представлення станів. В HTN-DL стан світу описується як база знань ДЛ. Це дозволяє використовувати дуже виразну мову представлення знань для представлення інформації про світ. Традиційне в ДЛ твердження відкритого світу (Open World Assumption) адоптується у міркування.

Встановлення відповідності задач планування та сервісів

У HTN-DL абстрактні сервіси представляються як задачі HTN-DL. В першу чергу, задача HTN-DL описується як концепт в онтології задач. Цей концепт представляє категорію сервісу. Це означає, що сервіси, які належать до цієї категорії, можуть забезпечувати бажану функціональність, та будь-який сервіс, що належить до цієї категорії, розглядаються як можливий кандидат. Звісно, ця категорія не повинна бути іменованим концептом, вона може бути складним концептом, який описує додаткові обмеження, які накладаються на сервіс. Ці обмеження типово визначаються для не функціональних атрибутів сервісів, тобто оцінювання, провайдер, атрибути якості і т. і.

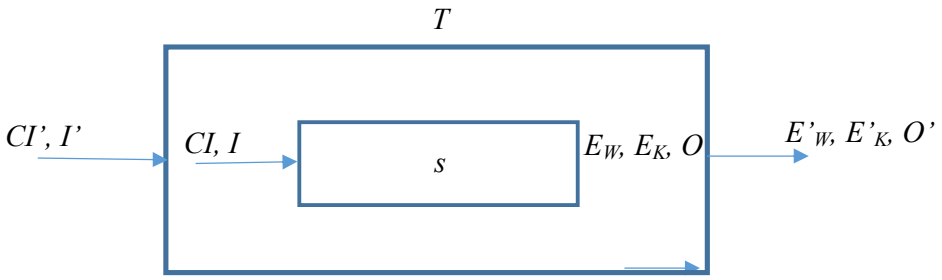
Окрім категорії сервісу, задачі можуть мати передумови та ефекти, що описують, коли сервіс може бути виконаний та що відбудеться після виконання. Але, коли задача співставляється з конкретним сервісом, передумови та ефекти не повинні співпадати повністю. У загальному випадку, відповідний сервіс може мати більш загальні передумови та більш специфічні ефекти [9]. Так як передумови та ефекти визначаються як кон'юнктивні ДЛ запити, відповідність може бути встановлена шляхом перевірки включення запитів.

Відповідно до визначення [3] HTN-задача задається шісткою

$$T = (N, I', O', CI', EW, EK),$$

де N – символ задачі, I – множина вхідних параметрів, O – множина вихідних параметрів, CI' – вираз передумови, $E'W$ – вирази ефектів та $E'K$ – ефекти знань, подібні до описів операторів. Таким чином, якщо сервіс описується кортежем $S = (I; CI; O, EW, EK)$, де I – список вхідних параметрів, CI – передумови сервісу, O – список вихідних параметрів, EW – ефекти сервісів, що змінюють світ, та EK – інформаційні ефекти сервісів, відповідність задачі та сервісу можна показати на рисунку, де

$$CI' \Rightarrow CI,\ E_W, E_K \Rightarrow E'_W, E'_K,\ I' \sqsupseteq I,\ O \sqsupseteq O'.$$



Рисунок

А задача встановлення відповідності між задачею та конкретним сервісом (задача виявлення сервісу) може бути визначена як автоматичне знаходження множини сервісів S таких, що

$$S = \{s \mid s = (I; CI; EW, EK; O),\ \ s \in R,\ CI' \Rightarrow CI,\ \ I \sqsubseteq I',\ \ EW, EK \Rightarrow E'W, E'K, O \sqsupseteq O'\}.$$

$\sqsubseteq$ означає відношення включення (subsumes), а $\Rightarrow$ – відношення імплікації.

Базові ступені відповідності, які використовуються механізмами співставлення, класифікуються відповідно до відношення включення наступним чином [10, 11]:

- *Exact* (точна), якщо задача та сервіс є еквівалентними концептами;
- *PlugIn*, якщо задача є підконцептом сервіс;
- Subsume, якщо задача є супер-концептом сервісу;
- Fail – в інших випадках, тобто відсутність відповідності.

У роботах [12, 13] можна знайти більш розширені класифікації, що використовують додаткові властивості відношення включення.

Основна відмінність співставлення задач у HTN-DL полягає в тому, що у критеріях співставлення розглядаються передумови та ефекти.

Після того, як встановлена відповідність та знайдені деякі сервіси кандидати, потрібно ще перевірити їх передумови, щоб впевнитися, що задовольнялися необхідні умови для використання цих сервісів.

Інформація про стан в HTN-DL також представляється за допомогою ДЛ-онтологій, тому обчислення передумов здійснюється як відповідь на ДЛ запит. Слід зазначити, що класичні системи планування типово адоптують міркування замкненого світу для обчислення передумов. Однак, завдяки стандартним ДЛ семантикам, в HTN-DL адоптуються міркування відкритого світу.

Деякі з відповідних сервісів можуть не мати ефектів в світі, але можуть надавати інформацію користувачеві. В HTN-DL, щоб зібрати інформацію, яка може використовуватися на наступних кроках вирішення задачі, інформаційні сервіси можуть виконуватися в ході планування.

HTN-DL та BPEL-процеси

Як видно з вищевикладеного, формалізм планування HTN-DL був розроблений саме для вирішення задач Веб-сервісів. Але для його застосування необхідно перекласти описи відповідних сервісів у HTN-DL представлення. В роботі [4] представлений транслятор для сервісів, описаних в OWL-S. Цей транслятор генерує HTN-DL домени з OWL-S описів. OWL-S було обрано як одну з досить поширених технологій семантичних Веб-сервісів. Але цей алгоритм трансляції демонструє, що управляючі потоки процесів Веб-сервісів можуть бути закодовані за допомогою методів HTN-DL, та даний формалізм може бути застосований і для сервісів, що мають інше представлення, зокрема й для BPEL сервісів.

Перш за все, слід зазначити, що BPEL призначений для опису складних інтегрованих процесів. Тому, даний засіб добре підходить для визначення процесної моделі складного сервісу. Але опис сервісу в BPEL проекті складається з трьох частин: WSDL-опис складного BPEL-процесу, BPEL-опису та XML – схем, які описують структури даних проекту. WSDL-опис визначає профіль сервісу. Стандартний опис профілю сервісу у WSDL лише визначає інтерфейс сервісу на двох рівнях: абстрактному та конкретному. На абстрактному рівні визначаються абстрактні операції, які може виконувати сервіс (при цьому під операцією розуміють простий обмін повідомленнями з клієнтом). Для операцій визначаються типи повідомлень та модель обміну. На конкретному рівні (рівні реалізації) елемент binding визначає точки доступу до сервісу (адреси у мережі та протоколи зв'язування), а також унікальне ім'я сервісу, що дозволяє створювати однозначні посилання на компоненти опису сервісу у відповідних сховищах. Таким чином, в стандартному WSDL – описі не охоплюються семантичні аспекти. Але WSDL – це звичайний XML, в якому передбачена можливість доповнення будь-якими елементами, які просто ігноруються системами, що не вміють їх читати. Ряд проектів,

що пропонуються для розгляду W3C, містять механізми для додаткового анотування конструкцій WSDL семантичними коментарями, які дозволяють зрозуміти сенс та функціональне призначення сервісу. Такими характеристиками одиниці функціональності можуть бути її складові, виконавці сервісу (програма або людина), статус (готовність), такі не функціональні характеристики, як якість сервісу, гарантії безпеки, тощо, а також shared variables – змінні, які є суттєвими для поведінки та використовуються у логічних формулах наступних характеристик:

1) перед-умови, які визначають інформаційний простір до виконання;
2) пост-умови, які визначають інформаційний простір після виконання;
3) припущення (assumptions), які визначають стан світу до виконання веб-сервісу;
4) ефекти, які визначають стан світу після виконання.

Всі умови, припущення та ефекти виражаються аксіомами, що представлені у мові формальної логіки. Це може бути, наприклад, мова WSMO або дескриптивна логіка.

Слід звернути увагу ще на де-які розбіжності представлення BPEL процесів та HTN-DL. Так, HTN-DL передбачає дві категорії сервісів. Це інформаційні сервіси та сервіси, що змінюють світ. Ефекти BPEL сервісів стандартно не описуються з такою категоризацією. Але ця проблема може бути вирішена просто анотуванням відповідних ефектів у WSDL як ефектів знань, як це робиться у HTN-DL.

Окрім цього, сервіс може мати умовні ефекти, які вказують, що зміни в світі відбудуться лише при виконанні певної умови. Це не є описом передумови, так як результат виконання дії в стані, де умова не задовольняється, має не визначений ефект. Умовні ефекти, з іншого боку, описують ефекти точно але при різних умовах можуть мати місце різні ефекти. HTN-DL не дозволяє умовних ефектів для методів та операторів, але цю виразність можна змоделювати додатковим методом. Наприклад, якщо оператор має умовний ефект, можна визначити метод та використовувати умови ефектів як умови мережі задач. У кожній мережі задач, можна використовувати окремий метод, що досягається трохи модифікованою версією оператора, тобто доповненої відповідним описом ефекту.

BPEL-опис задає процесну модель сервісу, тобто описує модель взаємодії з Web сервісом. Для кожного сервісу його профіль необхідно транслювати в елемент онтології задач, а модель процесу в набір методів та операторів. Перевагою BPEL-процесів є можливість опису як виконуваних, так і абстрактних процесів, тобто можливість представлення абстрактної функціональності.

Транслювання профілів сервісів, умов та ефектів

Профілі сервісів представляються як множина тверджень Abox дескриптивної логіки. Профіль сервісу є екземпляром конкретного концепту, який називається ServiceProfile. Описи профіля пов'язані з "параметрами сервісу", які описуються різні властивості сервісу. Параметри сервісу – це просто деякі відношення, які стверджуються у ABox. Ці відношення можуть стосуватися й багатьох нефункціональних питань, таких як, якість сервісу, наприклад, середній час відгуку, протокол безпеки, наприклад, алгоритм кодування, провайдер сервісу, наприклад, місце розміщення провайдера та його повноваження. Дескриптивна логіка та WSMO є онтологічними мовами, тому всі умови є просто їх аксіомами.

Онтології, що використовуються для опису сервісів, можуть також описувати ієрархію профілів. Ієрархія профілів є категоризацією сервісів, де зазвичай ServiceProfile є верхнім концептом. Категорії сервісів визначаються шляхом опису де-яких обмежень параметрів профілів сервісу. Якщо для опису ієрархії профілів сервісів використовувати онтології, то ці онтології можуть бути використані й як онтології задач HTN-DL. WSDL – описи профайлів сервісів фактично є OWL – онтологіями, тому вони можуть безпосередньо використовуватися як онтології задач HTN-DL.

Єдиним питанням залишається те, що екземпляри в онтології задач повинні бути відображені в існуючі визначення операторів та методів. Окрім цього, необхідно встановити відношення між операторами і методами, що є результатом трансляції процесних моделей, з екземплярами, які є результатом трансляції описів профайлів. Але як профайл, так і сам процес, належать сервісу, тому це відношення є тривіальним.

Перетворення моделей процесів. HTN-DL не підтримує конкурентних процесів. BPEL дозволяє як синхронну або асинхронну взаємодію з сервісами-партнерами (конструкція <partnerLinkType> у WSDL-описі), так і безпосередньо паралельне виконання дій всередині самого процесу (конструктор BPEL Flow). Але асинхронне виконання web сервісів досягається шляхом використання черги для обробки повідомлень, якими обмінюються ці сервіси. У HTN-DL дозволяється чергування композитних дій. Використовуючи цю властивість конкурентні дії можна транслювати у невпорядковану множину методів, які можуть чергуватися будь-яким чином. Хоча таке кодування виключає плани з паралельним виконанням атомних процесів, з точки зору семантики план залишається коректним, тому що немає жорстких обмежень на одночасне виконання.

Перетворення логічних умов. Логічні вирази в описах BPEL-процесів можуть бути виражені, у загальному випадку, в різних мовах, таких як Semantic Web Rule Language (SWRL) [14], Knowledge Interchange Format (KIF) [15], Declarative RDF System (DRS) [16, 17] або за допомогою Дескриптивної логіки [18]. Всі ці мови мають різні виразні можливості для опису різного типу формул, але, у загальному випадку, ці логічні вирази обмежені кон'юнкцією атомних фактів. Таким чином, в більшості випадків можна розглядати формулу

умови просто як список атомів. У такому разі, вона напряму відображається в умови HTN-DL, які мають аналогічну форму *AtomList*. Виключення становлять атоми, що містять операції для порівняння значень даних, виконання математичних обчислень, обробку строкових значень і т. і. Справа в тому, що такі операції не включені до HTN-DL формалізму. Було б доцільно додати відповідні розширення до HTN-DL, наприклад в стилі SHOP2, враховуючи що HTN-DL також є алгоритмом прямого планування, а умови оцінювання у стані надають значення для змінних, до яких могли б бути застосовані ці вбудовані функції.

Перетворення керуючих конструкцій. Кожна керуюча конструкція BPEL перетворюється у задачу HTN-DL та множину HTN-DL методів, що досягають цієї задачі. Керуючі конструкції BPEL складається з двох категорій діяльностей: базові та структуровані діяльності. Базові діяльності – це атомні дії, які включають конструкції receive, reply, assign, invoke, throw, terminate, empty та wait. Як і в мовах програмування, структуровані діяльності встановлюють обмеження залежностей керуючого потоків на виконання базових або структурованих діяльностей всередині них. Структурована діяльність може містити довільну глибину під-діяльностей. Структурові діяльності включають pick, switch, while, sequence, flow, scope, eventHandlers, faultHandlers, compensationHandler. Тоді, входами алгоритму трансляції є X – базова чи структурована BPEL конструкція та P – батьківський BPEL процес, який вміщує цю конструкцію. На виході необхідно отримати пару $< t, D >$, де t – опис задачі для X та D – опис домену, що містить всі оператори та методи, згенеровані для X. Для кожної діяльності BPEL потрібна окрема функція трансляції.

Визначення функцій трансляції, аналіз залежностей BPEL даних та розробка конкретних алгоритмів трансляції потоків даних та керуючих конструкцій є напрямком подальших досліджень.

Висновки

Сервіси, процедурна поведінка яких задана за допомогою BPEL, легко трансформуються в систему перехідних станів (STS). А більшість підходів планування покладаються на загальну модель, модель системи перехідних станів. Тобто, якщо Веб-сервіси транслювати у оператори планування, а ціль виразити логічною умовою, тоді можливо використати планувальник для генерації плану, який по суті є послідовністю екземплярів Веб-сервісів, а саме, послідовною композицією, що призводить до того, щоб цільова умова була вірною при їх виконанні. Але, пряме кодування композиції Веб-сервісів як задачі планування має декілька проблем. По-перше, звичайна логіка, що використовуються для представлення передумов та ефектів в системі планування має радикально іншу виразну потужність ніж мови Веб-онтологій. Ще одне питання – це Твердження Замкненого Світу, на яке спираються системи AI планування. Окрім цього, представлення Веб-сервісу як оператора планування має ще дві проблеми:

1) неможливо описати внутрішню структуру композитного сервісу;

2) оператор планування має лише інформацію про передумови та ефекти, але нам необхідно мати можливість виражати інформацію про самі сервіси, таку як провайдер сервісу, повноваження провайдера, оцінки сервісу користувачами та інші.

Комбінування дескриптивної логіки з існуючими підходами інтелектуального планування дозволяє подолати перелічені проблеми та дає ефективні механізми вирішення задач автоматизованої композиції Веб-сервісів на процесному рівні.

1. *Combining* Description Logic Reasoning with AI Planning for Composition of WEB Services // Evren Sirin, Doctor of Philosophy. – 2006.
2. *http*://ai-center.botik.ru/planning/index.php?ptl=book.htm
3. *Malik Ghallab, Dana Nau, and Paolo Traverso*. Automated Planning: Theory and Practice. Morgan Kaufman, San Francisco, CA, May 2004.
4. *Evren Sirin.* Combining Description Logic Reasoning with AI Planning for Composition of WEB Services, dissertation, Doctor of Philosophy, 2006.
5. *Premkumar T. Devanbu and Diane J. Litman.* Taxonomic plan reasoning. Artif. Intell. – 1996. – 84(1-2). – P. 1–35.
6. *Liviu Badea.* Planning in description logics: Deduction versus satisfiability testing. In Description Logics, 1998.
7. *Giuseppe De Giacomo, Luca Iocchi, Daniele Nardi, and Riccardo Rosati.* Description logic-based framework for planning with sensing actions. In Proceedings of the 1997 Description Logic Workshop (DL'97). – 1997. – P. 39–43.
8. *Luca Iocchi, Daniele Nardi, and Riccardo Rosati.* Planning with sensing, concurrency, and exogenous events: Logical framework and implementation. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR2000: Principles of Knowledge Representation and Reasoning, pages 678–689, San Francisco, 2000. Morgan Kaufmann.
9. *Srividya* Kona, Ajay Bansal, Gopal Gupta Department of Computer Science The University of Texas at Dallas Richardson, TX 75083, Thomas D. Hite Metallect Corp. 2400 Dallas Parkway Plano, TX 75093Automatic Composition of Semantic Web Services.
10. *Semantic* Matching of Web Service Capabilities. Massimo Paolucci, Takahiro Kawamora, Terry R. Payne, Katya Sycara.
11. *Semantic* Web Service Composition Based on a Closed World Assumption. Freddy L´ecu´ e, Alain L´eger, France Telecom R&D, France 4 rue du clos courtel, F-35512 Cesson S´evign´ e {(freddy.lecue, alain.leger)@orange-ft.com}, ´Ecole Nationale Sup´erieure des Mines de St-Etienne, France 158, cours Fauriel, F-42023 Saint-´Etienne, http://ieeexplore.ieee.org, 2006
12. *Javier Gonzalez-Castillo, David Trastour, and Claudio Bartolini.* Description Logics for Matchmaking of Services. In Workshop on Applications of Description Logics ADL 2001, Vienna, 2002.
13. *Lei Li and Ian Horrocks.* A Software Framework For Matchmaking Based on Semantic Web Technology. In Proc. of the Twelfth International World Wide Web Conference (WWW 2003), Budapest, Hungary, May 2003.
14. *Ian Horrocks and Peter F.* Patel-Schneider. A proposal for an owl rules language // In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004). ACM, 2004.

15. *Michael R. Geneserth and Richard E. Fikes.* Knowledge Interchange Format Version 3.0 Reference Manual. Technical Report Logic Group Report Logic-92-1 // Computer Science Department, Stanford University, June 1992.
16. *McDermott D.* Drs: A set of conventions for representing logical languages in rdf. http://www.daml.org/services/owl-s/1.0/DRSguide.pdf, 2004.
17. *McDermott D. and D. Dou.* Representing disjunction and quantifiers in RDF. In I. Horrocks and J. Hendler, editors, ISWC 2002, volume 2342, pages 250–263, 2002.
18. *Baader F. and Nutt W.*; In *Baader F., Calvanese D., McGuinness D., Nardi D., and Patel-Schneider P.* Basic Description Logics // The Description Logic Handbook, Cambridge University Press, 2003. – P. 43–95.

References

1. Combining Description Logic Reasoning with AI Planning for Composition of WEB Services. Evren Sirin, Doctor of Philosophy, 2006.
2. http://ai-center.botik.ru/planning/index.php?ptl=book.htm
3. Malik Ghallab, Dana Nau, and Paolo Traverso. Automated Planning: Theory and Practice. Morgan Kaufman, San Francisco, CA, May 2004.
4. Evren Sirin. Combining Description Logic Reasoning with AI Planning for Composition of WEB Services, dissertation, Doctor of Philosophy, 2006.
5. Premkumar T. Devanbu and Diane J. Litman. Taxonomic plan reasoning. Artif. Intell., 84(1-2):1–35, 1996.
6. Liviu Badea. Planning in description logics: Deduction versus satisfiability testing. In Description Logics, 1998.
7. Giuseppe De Giacomo, Luca Iocchi, Daniele Nardi, and Riccardo Rosati. Description logic-based framework for planning with sensing actions. In Proceedings of the 1997 Description Logic Workshop (DL'97), pages 39–43, 1997.
8. Luca Iocchi, Daniele Nardi, and Riccardo Rosati. Planning with sensing, concurrency, and exogenous events: Logical framework and implementation. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR2000: Principles of Knowledge Representation and Reasoning, pages 678–689, San Francisco, 2000. Morgan Kaufmann.
9. Srividya Kona, Ajay Bansal, Gopal Gupta Department of Computer Science The University of Texas at Dallas Richardson, TX 75083, Thomas D. Hite Metallect Corp. 2400 Dallas Parkway Plano, TX 75093Automatic Composition of Semantic Web Services.
10. Semantic Matching of Web Service Capabilities. Massimo Paolucci, Takahiro Kawamora, Terry R. Payne, Katya Sycara.
11. Semantic Web Service Composition Based on a Closed World Assumption. Freddy L´ecu´ e, Alain L´eger, France Telecom R&D, France 4 rue du clos courtel, F-35512 Cesson S´evign´ e {(freddy.lecue, alain.leger)@orange-ft.com}, ´Ecole Nationale Sup´erieure des Mines de St-Etienne, France 158, cours Fauriel, F-42023 Saint-´Etienne, http://ieeexplore.ieee.org, 2006
12. Javier Gonzalez-Castillo, David Trastour, and Claudio Bartolini. Description Logics for Matchmaking of Services. In Workshop on Applications of Description Logics ADL 2001, Vienna, 2002.
13. Lei Li and Ian Horrocks. A Software Framework For Matchmaking Based on Semantic Web Technology. In Proc. of the Twelfth International World Wide Web Conference (WWW 2003), Budapest, Hungary, May 2003.
14. Ian Horrocks and Peter F. Patel-Schneider. A proposal for an owl rules language. In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004). ACM, 2004.
15. Michael R. Geneserth and Richard E. Fikes. Knowledge Interchange Format Version 3.0 Reference Manual. Technical Report Logic Group Report Logic-92-1, Computer Science Department, Stanford University, June 1992.
16. D. McDermott. Drs: A set of conventions for representing logical languages in rdf. http://www.daml.org/services/owl-s/1.0/DRSguide.pdf, 2004.
17. D. McDermott and D. Dou. Representing disjunction and quantifiers in RDF. In I. Horrocks and J. Hendler, editors, ISWC 2002, volume 2342, pages 250–263, 2002.
18. F. Baader and W. Nutt; In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. Basic Description Logics / The Description Logic Handbook, Cambridge University Press, 2003. – P. 43–95.

Про автора:

Захарова Ольга Вікторівна,
кандидат технічних наук, старший науковий співробітник,
Кількість наукових публікацій в українських виданнях – 25.
http://orcid.org/0000-0002-9579-2973.

Місце роботи автора:

Інститут програмних систем НАН України,
03187, Київ, проспект Академіка Глушкова, 40.
Тел.: (068) 594 7560.
E-mail: ozakharova68@gmail.com.