

КОМПЛЕКСНЫЙ ПОДХОД К АВТОМАТИЗАЦИИ КОМПОЗИЦИИ ВЕБ-СЕРВИСОВ

Проведено аналіз проблеми автоматизації композиції веб-сервісів. Запропоновано критерії аналізу підходів до автоматизації композиції веб-сервісів. На основі запропонованих критеріїв розроблено підхід, що передбачає виконання формальної специфікації композиції на мові темпоральної логіки TLA та акцентується на аспектах верифікації, валідації та планування.

The analysis of Web Service Composition automation challenge has been conducted. The criterions for the analysis of approaches to Web Service Composition automation have been proposed. The approach based on these criterions has been developed. The approach envisages composition formal specification by using TLA temporal logic and covers the aspects of verification, validation and planning.

Введение

Сервис-ориентированная архитектура (*Service Oriented Architecture, SOA*) – компонентная модель, структурными элементами которой являются веб-сервисы. Парадигма *SOA* предполагает повторное использование уже созданных программных компонентов с целью расширения набора доступных функциональных возможностей. За счет этого уменьшается избыточность и затраты, связанные с разработкой соответствующих программных компонентов.

Развитием идеи повторного использования программных компонентов, представленных в виде веб-сервисов (далее – сервисов), является концепция композиции, которая заключается в формировании композитных сервисов с новыми функциональными возможностями на основе уже имеющихся сервисов. Будем далее различать композитные сервисы – сервисы, созданные из других сервисов, и атомарные сервисы – сервисы, из которых создан композитный сервис. Таким образом, атомарный сервис также может представлять композитный сервис, если он входит как составная часть в более сложный композитный сервис. Под композицией сервисов будем подразумевать процесс создания композитного сервиса.

Композиция сервисов связана с решением многих задач, в частности, разработкой методов согласования атомарных сервисов, моделированием и практической реализацией композитных сервисов, организацией их функционирования, верификацией и валидацией [1]. Неко-

торые стандарты, посвященные композиции сервисов, например, *WS-BPEL (Web Service Business Process Execution Language)* и *WS-CDL (Web Service Choreography Description Language)*, описывают механизмы композиции сервисов, но не учитывают вопросы, связанные с автоматизацией создания композитных сервисов, возникающие при их практической реализации [2]. В настоящее время уже имеется несколько подходов к автоматизации создания композитных сервисов, вот некоторые из них [3, 4]:

- 1) композиция на основе ситуационного исчисления (с использованием языка логического программирования *Golog*);
- 2) композиция с учетом семантики сервисов (*OWL-S, WSMO, PDDL*);
- 3) планирование композитных сервисов на основе ролей, при котором композитный сервис специфицируется на языке *Composite Service Specification Language (CSSL)* и, затем, выполняется согласование ролей атомарных сервисов для последующей генерации композитного сервиса;
- 4) синтез сервисов с применением методов доказательства теорем к спецификациям сервисов на языках *OWL-S, SSP* и др.

1. Постановка задачи

Как показывает анализ существующих подходов к композиции сервисов, они не носят комплексный характер, т.е. не учитывают всех составляющих процесса синтеза сложных сис-

тем: от спецификации требований заказчика сервиса – до получения конечного результата сервиса или, как принято говорить, "end-to-end". Учитывая, что композитный сервис может потребоваться заказчику в динамическом окружении и в режиме "ad hoc", большая часть процесса композиции должна быть автоматизированной.

В данной работе ставится задача проанализировать современные исследования в области композиции сервисов, обосновать и структурировать (стратифицировать) проблему автоматизации композиции сервисов и предложить пути ее решения.

2. Составляющие проблемы композиции

В контексте последовательного процесса "end-to-end" – от момента поступления клиентского запроса на определенную веб-услугу – до получения клиентом результатов работы композитного сервиса [5] – проблема автоматизации композиции сервисов носит комплексный характер и включает несколько взаимосвязанных задач (рис. 1).

Автоматизация композиции требует, прежде всего, однозначной машинной интерпретации клиентского запроса, атомарных сервисов и композитного сервиса. Для этого необходимо найти соответствующие способы и формы спецификации сервисов. Наиболее часто для машинно-ориентированной спецификации сервисов и распределенных компьютерных процессов применяют следующие способы спецификации: спецификации на графических языках (*UML*, *SysML*); спецификации в виде текста на

диалектах языка *XML* (*WSDL*, *UDDI*, *SOAP*, *WSDP*); формальные спецификации с использованием темпоральных логик (*LTl*, *CTL*) [6]. На наш взгляд, формальная спецификация является наиболее приемлемой в контексте автоматизации композиции сервисов.

Спецификация клиентского запроса должна включать требования к функциональным возможностям запрашиваемого сервиса и может содержать требования к качеству сервиса (*Quality of Services*, *QoS*), т.е. нефункциональные требования. Учет *QoS* может сыграть определяющую роль при планировании композиции, когда все функциональные требования уже выполнены, позволяя проводить оптимизацию за счет сокращения набора рассматриваемых экземпляров сервисов. Например, если в качестве параметра *QoS* выступает время отклика сервиса, то композиция сервисов может быть интерпретирована как задача о рюкзаке и клиенту предоставляется композитный сервис с наименьшим аккумулярованным значением *QoS* [7]. Характеристиками *QoS* могут быть также пропускная способность сетевого канала, стоимость отдельного сервиса, его доступность или готовность и т.п. Альтернативные подходы к композиции, предполагающие спецификацию нефункциональных параметров, предложены в работах [8, 9].

Отметим, что на конкурсе *Web Service Challenge* (*WSC* 2010), организованном Т. Вейсом [10], ставилась задача разработки эффективных средств автоматизированной композиции веб-сервисов с развитыми механизмами планирования. В предложенной инициативе предлагается выполнять проектирование средств автоматизации композиции сервисов с использованием языков *WSDL*, *OWL-S*, *WS-BPEL*, а в качестве нефункциональных требований рассматривать время отклика атомарного сервиса и пропускную способность сетевого канала. Последнее требование не достаточно обоснованно, т.к. в настоящее время современные сетевые средства уже обеспечивают возможность динамической композиции сервисов, а скорость взаимодействия распределенных процессов с каждым годом возрастает. На наш взгляд, большие накладные расходы имеют алгоритмы практического планирования композиции сервисов.

Анализу эффективности алгоритмов планирования композиции посвящены работы [11, 12], в которых акцент ставится на производительности, а реализация основывается на ис-



Рис. 1. Стратификация проблемы автоматизированной композиции сервисов

пользовании иерархических сетей *HTNs (Hierarchical Task Networks)*. К недостаткам этих алгоритмов можно отнести зависимость от предметной области (ПрО) и детерминизм структуры композиции [13]. Возможное решение может заключаться в использовании генетических алгоритмов, однако они требуют для некоторых задач значительных затрат времени поиска решений.

Для проверки правильности формальной спецификации и корректности функционирования композитного сервиса используются, соответственно, процедуры верификации и валидации [6]. Но, следует отметить, что вопросы формальной верификации и, особенно, валидации сервисов еще не достаточно освещены в публикациях, а средства для их проведения еще не получили всеобщего признания. Основными методами верификации и валидации является инспектирование спецификаций и моделирование вариантов использования системы (use-cases).

3. Модели композиции

Существующие модели композиции сервисов условно можно разделить на два класса: статические и динамические [4], которые базируются на соответствующих стандартах (рис. 2).

Под статической композицией понимается процесс, при котором композитный сервис создается на этапе его разработки. Динамическая композиция выполняется после запроса клиентом определенных услуг композитного сервиса и может допускать внесение некоторых изме-

нений в его функциональные возможности. Модели динамической композиции обладают адаптивностью к изменениям внешних условий, изменениям требований клиента, статусу и составу атомарных сервисов, рассматривающихся в качестве потенциальных кандидатов на включение в композитный сервис. Трудно предложить единый подход для реализации динамических моделей из-за огромного диапазона возможных целевых функций и предметных областей композитных сервисов. Установить общность свойств динамических композиций будет возможно по мере увеличения степени абстрагирования бизнес-активностей от конкретных предметных областей, развития метамodelей композитных сервисов и разработки новых стандартов композиции.

Наиболее распространенным на сегодняшний день стандартом композиции сервисов является стандарт *WS-BPEL*, который поддерживается консорциумом *OASIS* [14]. На его основе разработаны многие прикладные программные системы. Общей характеристикой этих систем является их корпоративная привязанность, что обуславливается целями, приоритетами и характером деятельности отдельных организаций, выступающих в роли источников финансирования разработки соответствующих программных систем, что, иногда, является причиной ухудшения интероперабельности сервисов. Стандарт *WS-BPEL* реализует модель оркестровки, основанную на асинхронном вызове атомарных сервисов в рамках абстрактного бизнес-процесса, который выполняет функцию координатора процесса композиции и оперирует *WSDL*-описаниями атомарных сервисов. Таким образом, бизнес-процесс – это, своего рода, верхний абстрактный уровень в контексте модели оркестровки.

Модель оркестровки обладает свойством адаптации к изменениям требований со стороны корпоративного сегмента, но имеет ряд недостатков:

- 1) модель не содержит механизмы автоматизации композиции;
- 2) модель не предусматривает верификацию и валидацию композитных сервисов;
- 3) модель допускает композицию только на этапе разработки.

Альтернативный и более высокоуровневый путь решения проблемы композиции сервисов предлагает модель хореографии на основе стандарта *WS-CDL* консорциума *W3C* [15], ко-



Рис. 2. Модели композиции

торый так и не получил статус окончательного стандарта. Первоначально спецификация процесса хореографии заключалась в описании механизмов взаимодействия между атомарными сервисами. В последствии она была дополнена положениями о взаимодействии между бизнес-процессами. Это можно обосновать более активным восприятием модели оркестровки со стороны корпоративного сегмента.

4. Комплексный подход к композиции и его обоснование

Согласно [13] можно выделить следующие требования, которые необходимо учитывать при разработке подходов к автоматизации композиции сервисов:

- 1) независимость от предметной области;
- 2) возможность адаптации к динамически меняющимся внешним условиям;
- 3) учет *QoS* требований к композитным сервисам;
- 4) наличие механизмов верификации и валидации;
- 5) возможность обработки ошибок.

С учетом этих требований, мы предлагаем следующие критерии оценки подходов к автоматизированной композиции сервисов:

- 1) способ спецификации (графическая, текстовая, формальная);
- 2) полнота обеспечения верификации и валидации (*V&V*) композитных сервисов;
- 3) учет нефункциональных (*QoS*) требований;
- 4) практическая применимость.

Способ спецификации дает возможность определить, предполагает ли подход формальную спецификацию композитных сервисов и, как следствие, автоматизацию процессов *V&V*. Возможность автоматизированной *V&V* формальной спецификации необходима для проверки корректности композитных сервисов и их характеристик.

Учет требований *QoS* к композитным сервисам позволит повысить качество предоставляемой услуги: параметром *QoS*, например, может служить время отклика экземпляра веб-сервиса [7].

Практическая применимость определяет эффективность подходов в случае участия в процессе композиции сотен экземпляров веб-сервисов. Большинство подходов, однако, обеспечивают автоматизированное создание композитных сервисов лишь из нескольких (до пяти) атомарных сервисов [10], что затрудняет

возможности удовлетворить требование независимости от предметной области.

Таким образом, комплексный подход к автоматизированной композиции сервисов должен содержать следующие элементы:

- 1) формальную спецификацию композитного (целевого) сервиса;
- 2) методы автоматизированной верификации и валидации композитных сервисов;
- 3) планирование процесса композиции.

В качестве языка формальной спецификации предлагается использовать язык темпоральной логики *TLA* (*Temporal Logic of Actions*), предложенной Л. Лампортом [16]. Преимущество *TLA*, по сравнению с логиками *LTL* (*Linear Temporal Logic*) и *CTL* (*Computation Tree Logic*), заключается в большей выразительности спецификации систем, состоящих из параллельных взаимодействующих процессов. В *TLA* вводится понятие "*behavior*", что позволяет описать каждый из возможных путей протекания отдельного процесса. Кроме того, для *TLA* имеются развитые инструментальные средства формальной спецификации и верификации, в частности, *TLA Toolbox*, использующий метод "*Model Checking*".

Учитывая, что композитные сервисы – это компьютерные распределенные системы взаимодействующих процессов, для решения задачи валидации необходимо использовать метод дискретно-событийного моделирования. Для этого можно, например, использовать программный комплекс *DEV Suite* [17]. Данный комплекс соответствует природе рассматриваемой проблемной области и имеет средства визуализации статистических данных.

Говоря об автоматизированном планировании композиции, отметим, что большинство существующих подходов построено либо на основе иерархических сетей (*HTNs*), либо с использованием генетических алгоритмов. Пока только они способны решить задачи планирования с участием в композиции сотен экземпляров сервисов за приемлемое время. При этом генетические алгоритмы зачастую требуют больших временных затрат по сравнению с *HTNs*, а *HTNs*, в свою очередь, привязаны к ПрО и требуют трансформации семантики ПрО в виде онтологии на языке *OWL-S* в описание на основе *HTNs* (планировщик *HTNWSC-P*). Поэтому, на наш взгляд, более перспективным для задач планирования процесса композиции с участием сотен экземпляров атомарных серви-

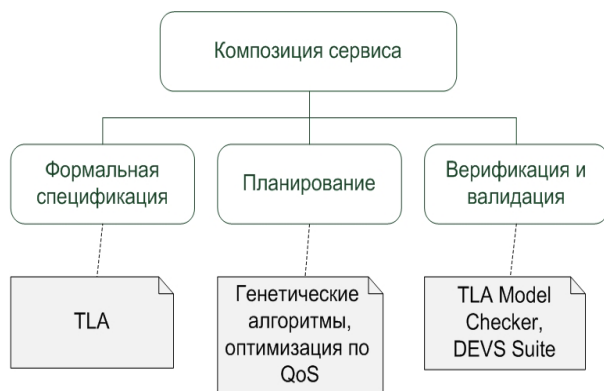


Рис. 3. Комплексный подход к композиции сервисов

сов является использование генетических алгоритмов.

Оценку эффективности работы алгоритма планирования композиции (критерий практической применимости) предполагается осуществлять на основе методов агентного имитационного моделирования, учитывая сходство рассматриваемой проблемной области и специфики агентного подхода, с помощью которого можно исследовать сложные децентрализованные и распределенные системы, состоящие из множества атомарных сервисов (агентов). Для имитационного моделирования таких систем можно использовать программные средства *SWARM* и *AnyLogic*.

Таким образом, структура предлагаемого подхода состоит из формальной спецификации, планирования, верификации и валидации (рис. 3).

5. Пример использования DEVS-формализма

Модель атомарной системы (в нашем случае атомарного сервиса) может быть описана с помощью DEVS-формализма (*Discrete Event System Specification*) в виде кортежа:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle, \quad (1)$$

где X – набор входных параметров; S – набор состояний системы; Y – набор выходных параметров; $\delta_{int} : S \rightarrow S$ – внутренняя функция переходов; $\delta_{ext} : Q \times X \rightarrow S$ – внешняя функция переходов; $Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$ – полный набор переходов; e – время, прошедшее с момента последнего перехода; $\lambda : S \rightarrow Y$ – функция выхода; ta – функция продвижения времени.

Рассмотрим простой пример композитного сервиса, состоящего из сервисов $ws1$, $ws2$ и $ws3$, развернутых, соответственно, на серверах (узлах) $s1, s2, s3$, причем, на серверах $s2$ и $s3$ реализован некоторый веб-портал. Потребителю композитного сервиса необходимо через сервис $ws1$ определить, соответствует ли значение пропускной способности сетевого канала, образуемого узлами $s1, s2, s3$, ожидаемому значению. Взаимодействие между сервисами можно отобразить в виде графа $G(V, E)$ (рис. 4) с вершинами $V = (ws1, ws2, ws3)$ и ребрами $E = (e12, e23, e31)$.

Алгоритм взаимодействия сервисов может быть таким:

- 1) потребитель композитного сервиса (инициатор транзакции) обращается к сервису $ws1$ с запросом определить пропускную способность между узлами $s1, s2$, передавая в качестве аргумента строку со списком *URI* (*Universal Resource Identifier*) узлов $s1, s2, s3$, причем, первый и последний элементы списка – *URI* узла $s1$;
- 2) сервис $ws1$ выполняет запрос к утилите *iperf* (<http://sourceforge.net/projects/iperf/>), с помощью которой определяет пропускную способность сетевого канала между узлами $s1, s2$;
- 3) сервис $ws1$ удаляет из указанного списка собственный *URI* и отправляет модифицированный список и вычисленное значение пропускной способности сервису $ws2$;
- 4) сервис $ws2$ выполняет аналогичные шагу 3 действия, вычисляет минимальное значение пропускной способности $\min(e12, e23)$. Это значение вместе с вновь модифицированным списком *URI* отправляется сервису $ws3$;
- 5) сервис $ws3$ также выполняет действия, аналогичные действиям на шаге 3, и отправляет сервису $ws1$ искомое значение пропускной способности сетевого канала.

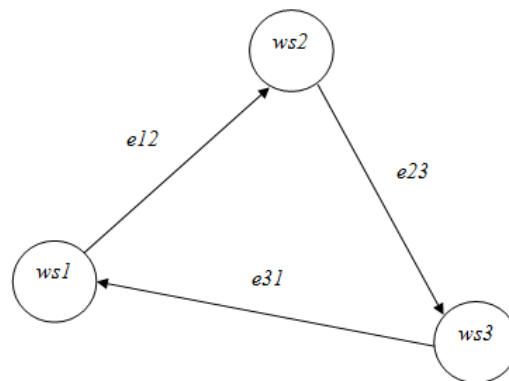


Рис. 4. Граф взаимодействия сервисов

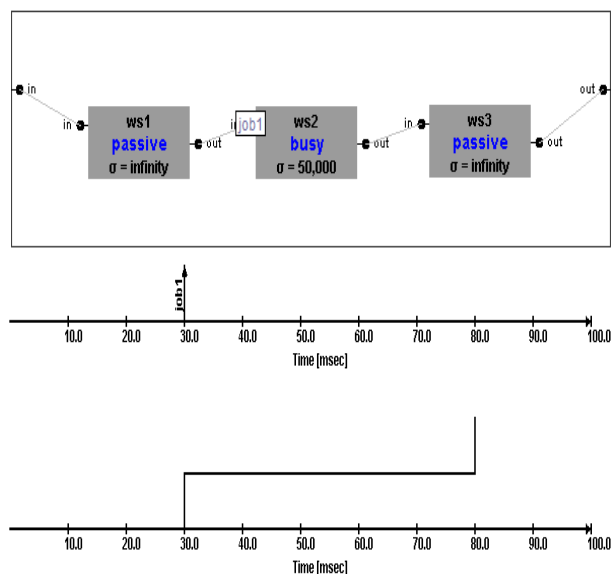


Рис. 5. Результат моделирования композитного сервиса в среде *DEVs Suite*

Смоделируем влияние значения параметра *QoS* (время отклика) отдельного атомарного сервиса (*ws2*) на процесс композиции с помощью инструментария *DEVs Suite*. С целью демонстрации лишь основных свойств взаимодействия сервисов не будем учитывать функциональные составляющие δ_{int} , δ_{ext} и λ , т.е. модели каждого из сервисов *ws1*, *ws2* и *ws3* имеют вид четырехместного кортежа:

$$M = \langle X, S, Y, ta \rangle, \quad (2)$$

где $S = \{ "busy", "passive" \}$.

Построенная имитационная модель композитного сервиса дает нам возможность проверить соответствие сервиса функциональным и нефункциональным требованиям запроса клиента. Удовлетворение функциональных требований подтверждается благодаря правильному функционированию модели, а нефункциональных – возможностью определять значения временных задержек, вносимых атомарными сервисами, что является необходимым условием валидации функционирования композитного сервиса.

На рис. 5 представлено графическое окно вывода *DEVs Suite*, полученное при моделировании процесса взаимодействия сервисов *ws1*, *ws2*, *ws3* в составе композитного сервиса.

Выводы

Таким образом, в данной статье проведен анализ проблемы автоматизации композиции веб-сервисов. Показано, что современные подходы к композиции не достаточно учитывают

необходимость верификации и валидации композитных сервисов.

Предложены критерии анализа подходов к автоматизации композиции веб-сервисов. На основе этих критериев разработан подход, предполагающий выполнение формальной спецификации на языке темпоральной логики *TLA*, включающий процессы верификации, валидации и планирования композиции.

Выполнено обоснование возможности применения агентного подхода к имитационному моделированию с целью оценки эффективности планирования процесса композиции.

Приведен пример моделирования взаимодействия трех атомарных сервисов в составе композитного сервиса, демонстрирующий возможности использования *DEVs*-формализма и программного пакета *DEVs Suite* для валидации функционирования композитных сервисов.

СПИСОК ЛИТЕРАТУРЫ

1. *Srivastava B.* Web Service Composition – Current Solutions and Open Problems / B. Srivastava, J. Koehler // *Proceedings of The 13th International Conference on Automated Planning & Scheduling* (Trento, Italy, June 10th, 2003). – P. 28 – 35.
2. *Дерецкий В. А.* Подход к композиции веб-сервисов на основе спецификации функциональной семантики / В. А. Деревский // *Проблемы программирования*. – 2009. – № 2. – С. 30 – 39.
3. *Rao J. A* Survey of Automated Web Service Composition Methods / J. Rao, X. Su // *Proceedings of The First International Workshop on Semantic Web Services* (San Diego, California, USA, July 6th, 2004). – P. 43 – 54.
4. *Bucchiarone A.* A Survey on Services Composition Languages and Models / A. Bucchiarone, S. Gnesi // *Proceedings of The International Workshop on Web Services – Modeling and Testing* (Palermo, Italy, June 9th, 2006). – P. 51 – 63.
5. *Agarwal V.* Synthy: A System for End to End Composition of Web Services / V. Agarwal, B. Srivastava // *Journal of Web Semantics*. – 2005. – Vol. 3, No. 6. – P. 311 – 339.
6. *Foster H.* Using a Rigorous Approach for Engineering Web Service Compositions: A Case Study / H. Foster, S. Uchitel, J. Magee, J. Kramer, M. Hu // *Proceedings of the IEEE International Conference on Service Computing* (Orlando, Florida, USA, July 11 – 15, 2005). – P. 217 – 224.
7. *Bin X.* Efficient Composition of Semantic Web Services with End-to-End QoS Optimization / X. Bin, L. Sen, Y. Yixin // *Tsinghua Science and*

- Technology. – 2010. – Vol. 15, No. 6. – P. 678 – 686.
8. Khan F. H. QoS Based Dynamic Web Services Composition & Execution / F. H. Khan, M. Y. Javed, S. Bashir, A. Khan, H. Khiyal // International Journal of Computer Science and Information Security. – 2010. – Vol. 7, No. 2. – P. 147 – 152.
9. Kritikos K. Semantic QoS-based Web Service Discovery Algorithms for Over-Constrained Demands / K. Kritikos, D. Plexousakis // International Journal of Web Services Practices. – 2008. – Vol. 3, No. 1-2. – P. 71 – 82.
10. Web Service Challenge 2010 [электронный ресурс]. – Режим доступа : \www/ URL: http://www.it-weise.de/documents/files/W2010WSC_pres.pdf. – Загл. с экрана.
11. Sohrabi S. Customizing the Composition of Actions, Programs and Web Services with User Preferences / S. Sohrabi // Proceedings of the 9th International Semantic Web Conference (Shanghai, China, November 7 – 11, 2010). – P. 321 – 329.
12. Sohrabi S. Preference-based Web Service Composition: A Middle Ground Between Execution and Search / S. Sohrabi, S. A. McIlraith // Proceedings of the 9th International Semantic Web Conference (Shanghai, China, November 7 – 11, 2010). – P. 713 – 729.
13. Medjahed B. Service Composition for the Semantic Web / B. Medjahed, A. Bouguettaya. – NY. : Springer, 2011. 191 p. – ISBN 978-1-4419-8464-7.
14. Web Services Business Process Execution Language Version 2.0 [Электронный ресурс] // Режим доступа: \www/ URL: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>
15. Web Services Choreography Description Language Version 1.0 [Электронный ресурс] // Режим доступа: \www/ URL: <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>
16. Chaudhuri K. The TLA+ Proof System: Building a Heterogeneous Verification Platform / K. Chaudhuri, D. Doligez, L. Lamport, S. Merz // Proceedings of the 7th International colloquium conference on Theoretical aspects of computing (Rio Grande do Norte, Brazil, September 1 – 3, 2010). – p. 44.
17. Cristia M. A TLA+ Encoding of DEVS Models / M. Cristia // Proceedings of International Modeling and Simulation Multiconference (Buenos Aires, Argentina, February 8 – 12, 2007). – P. 17 – 22.

V. V. Shkarupilo, R. K. Kudermetov. **The comprehensive approach to Web Service Composition automation.**

В. В. Шкарупило, Р. К. Кудерметов. **Комплексний підхід до автоматизації композиції веб-сервісів.**

Проведен анализ проблемы автоматизации композиции сервисов. Предложены критерии анализа подходов к автоматизации композиции сервисов. На основе предложенных критериев разработан подход, предполагающий выполнение формальной спецификации композиции на языке темпоральной логики TLA и акцентирующийся на аспектах верификации, валидации и планирования.