

УДК 004.94

В.В. Шкарупило, С.Ю. Скрупский, Р.К. Кудерметов  
Запорожский национальный технический университет

## DEVS-МОДЕЛЬ КАК СРЕДСТВО ВАЛИДАЦИИ КОМПОЗИТНЫХ ВЕБ-СЕРВИСОВ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

*Виконано обґрунтування придатності DEVS-формалізму (Discrete Event System Specification) та його реалізації у середовищі дискретно-подійного імітаційного моделювання DEVS Suite для валідації композитних веб-сервісів. З цією метою було проведено порівняльний аналіз середовищ моделювання DEVS Suite, GPSS, AnyLogic, GridSim, MATLAB, а також створено комп'ютерні моделі композитного веб-сервісу у середовищах DEVS Suite та GPSS.*

Ключові слова: SOA, композитний веб-сервіс, валідація, DEVS, GPSS, розподілена система.

### Введение

Сервис-ориентированная архитектура (Service Oriented Architecture, SOA) – это компонентная модель, элементами которой являются веб-сервисы (Web Services, WS), взаимодействующие в компьютерной среде, в общем случае, с использованием компьютерной сети. Ключевым свойством SOA является интероперабельность, т.е. способность WS взаимодействовать между собой. Интероперабельность достигается за счет соответствующих стандартов, разработанных организациями W3C, OASIS и WS-I. Интерфейсы WS специфицируются с помощью языка Web Service Description Language (WSDL). Спецификации WS содержат описания портов, типы сообщений, реализуемые WS функции и другие данные, необходимые для взаимодействия WS. Информация о WS может публиковаться в реестрах, например UDDI, и, таким образом, быть доступной для использования потенциальными клиентами.

Еще одной особенностью архитектуры SOA является возможность генерации новых, композитных WS (Composite Web Service, CWS) из других WS, которые называются атомарными (Atomic Web Services, AWS). Процессы формирования CWS и организации взаимодействия AWS в составе CWS получили названия "оркестровка" и "хореография" WS (Web Services Orchestration & Choreography) [1, 2]. Таким образом, WS могут синтезироваться, модифицироваться, быть распределенными во времени и пространстве и принадлежать разным провайдерам.

Одной из задач автоматизации процесса композиции CWS является их валідація. Она заключается в проверке соответствия услуг (сервисов), предоставляемых CWS, требованиям потребителей этих услуг. Такая проверка на этапе синтеза CWS осуществляется при помощи методов моделирования.

### 1. Постановка задачи

Наиболее подходящим, на наш взгляд, методом моделирования CWS с целью валідації является дискретно-событийный имитационный метод моделирования. Существуют различные программные среды для организации дискретно-событийного моделирования систем, в частности DEVS Suite, которая основана на DEVS-формализме [3].

В данной работе ставится задача проанализировать и подтвердить возможности использования моделирующей среды DEVS Suite и DEVS-формализма для валідації CWS. Следует отметить, что выбор этих средств в качестве инструментария валідації CWS обусловлен: во-первых, математической строгостью DEVS-формализма, который является расширением конечного автомата Мили; во-вторых, на базе этого формализма уже разработан параллельный DEVS-формализм, предназначенный для моделирования классов параллельных и распределенных взаимодействующих процессов, к которым относятся CWS; в-третьих, в настоящее время на основе DEVS-формализма развиваются моделирующие среды, использующие концепцию SOA [4], что позволит интегрировать процедуру валідації в процесс композиции CWS в автоматизированном режиме.

### 2. Анализ сред моделирования

Результаты сравнительного анализа сред моделирования DEVS Suite, GPSS, AnyLogic, GridSim и MATLAB с точки зрения наибольшего соответствия задаче валидации CWS приведены в табл. 1. Для задания критериев, по которым сравнивались среды моделирования, использовался список вопросов, приведенный в [5], а также возможности программной реализации имитационных моделей (ИМ) функционирования WS в этих средах.

Таблица 1

Оценка сред моделирования в качестве средств валидации CWS

Критерии	Среды моделирования				
	DEVS Suite	GPSS	AnyLogic	GridSim	MATLAB
Возможность наблюдения за изменением значений параметров компонентов ИМ	+	+	+	+	+
Допустимость воздействия на ход моделирования	+	-	+	+	+
Удобство внесения изменений в ИМ	+	+	+	+/-	+/-
Соответствие исследуемой предметной области	+	+/-	+/-	+/-	-
Легкость описания ИМ	+/-	+	+	+/-	+/-
Удобство и наглядность отчетов о процессе моделирования	+/-	+	+	+/-	+

"+" – среда моделирования удовлетворяет заданному критерию;

"-" – не удовлетворяет;

"+/-" – частично удовлетворяет.

Ключевым аргументом в пользу DEVS Suite является наибольшее соответствие концепций SOA и DEVS-формализма, а именно, поведение AWS может описываться отдельными (атомарными) моделями, а CWS – композитными моделями. DEVS Suite позволяет программировать алгоритмы функционирования WS, например, на языке Java.

GPSS – хорошо зарекомендовавший себя продукт, на протяжении более чем 40 лет являющийся стандартом де-факто в области имитационного моделирования [6].

AnyLogic относится к числу стремительно развивающихся сред моделирования. Особенность данного инструментария заключается в агрегировании следующих методов имитационного моделирования: системная динамика, дискретно-событийное моделирование, агентное моделирование. Среда AnyLogic достаточно универсальна, ориентирована на коммерческое использование, имеет развитый интерфейс пользователя и высокую степень автоматизации процесса моделирования [7].

GridSim представляет собой среду моделирования, созданную для исследования функционирования и взаимодействий между Grid-сервисами [8].

Несмотря на универсальность и мощь инструментальной среды MATLAB [9], она мало подходит для целей валидации CWS, т.к. обладает ограниченными возможностями программирования WS.

Авторы проанализировали и другие среды имитационного моделирования, в частности, приведенные в обзоре [10]. Исходя из того, что характер реализации имитационной (компьютерной) модели исследуемой системы определяется спецификой выбранной среды моделирования, концепция атомарных моделей, положенная в основу DEVS-формализма, обеспечивает естественный механизм отображения функциональных и нефункциональных характеристик AWS в соответствующие атомарные DEVS-модели в составе модели CWS. Учитывая открытость кода и доступность среды DEVS Suite, все это послужило обоснованием выбора инструментария DEVS Suite в качестве среды моделирования для решения задач валидации.

### 3. Пример системы с CWS и ее моделирование в среде DEVS Suite

В качестве примера системы, в которой используется CWS, рассмотрим систему распределенной обработки данных.

Предположим, что каждый из компонентов системы представляет собой отдельный аппаратный узел, на котором развернут WS с заданными функциональными характеристиками. Пусть взаимодействие с CWS осуществляется по схеме "клиент-сервер" через компьютерную сеть. При этом клиент может быть представлен как AWS, который генерирует запросы к CWS на обработку блоков данных. Предположим также, что CWS – это совокупность распределенных AWS. Сценарий распределенной обработки данных следующий:

- клиент формирует запросы, в которых описываются задания на выполнение определенных вычислений и специфицируются требования к характеристикам вычислительных узлов (ВУ);
- запросы клиента направляются планировщику, который осуществляет поиск подходящих ВУ и назначает им задания, при этом объем вычислений для каждого ВУ предполагается одинаковым;
- ВУ выполняют вычисления и полученные результаты отправляют в хранилище, откуда последние могут быть получены клиентом.

В табл. 2 приведены компоненты системы, которые учитываются в модели CWS, их обозначения и назначение. Для моделирования временных задержек на обмен данными в коммуникационной среде в модель введены блоки "Коммуникатор". Структурная схема модели системы приведена на рис. 1.

Таблица 2

Компоненты системы с CWS

Компонент	Обозначение в модели	Назначение
Клиент	Client_svc	Генерация заявок на обслуживание (job_i).
Планировщик	Planner_svc	Назначение job_i на ВУ.
ВУ (2 шт.)	WN_0_svc, WN_1_svc	Обработка job_i.
Хранилище	Storage_svc	Хранение результатов обработки job_i.
Коммуникатор	Net_xx	Имитация временных задержек при передаче данных по сетевым каналам.

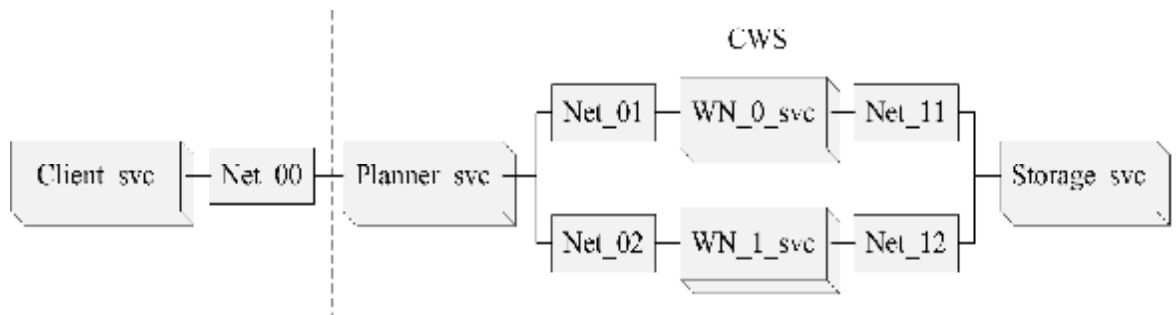


Рис. 1. Структурная схема моделируемой системы

DEVS-формализм предоставляет возможность специфицировать каждый из компонентов моделируемой системы в форме атомарной модели. Каждый из компонентов системы может быть описан с помощью DEVS-формализма в виде кортежа:

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

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

$\lambda : S \rightarrow Y$  – функция выхода;

$ta$  – функция продвижения времени.

Функция  $ta$  задает текущий и последующий моменты времени изменения состояния  $s$  атомарного компонента модели. Разница между этими моментами выражается через значение  $s$ .

Компоненты системы могут принимать следующие возможные состояния:

- "passive" – компонент простаивает;
- "busy" – компонент загружен (занят).

Изменение состояния системы происходит в случае изменения состояния одного из ее компонентов.

Внешняя функция перехода  $d_{ext}$  задает функции атомарных компонент модели. Описание этих функций приведено в табл. 3.

Таблица 3

Функции компонентов DEVS-модели

Атомарный компонент	Функция перехода, $d_{ext}$
Client_svc	Генерация заявок на обслуживание (job_i).
Planner_svc	Прием job_i в очередь и определение ВУ (WN_0_svc, WN_1_svc) с состоянием "passive", куда будет отправлена на обработку job_i.
WN_0_svc, WN_1_svc	Обработка job_i.
Storage_svc	Хранение результатов обработки job_i.
Net_00, Net_01, Net_02, Net_11, Net_12	Имитация временной задержки в сетевом канале (канале связи).

В табл. 4 приведены временные задержки на компонентах DEVS-модели, которые приняты из следующих соображений:

- временная задержка для блока Client\_svc задана искусственно. Считаем, что заявки job\_i генерируются равномерно через каждые 3 с;

- время, затрачиваемое на решение задачи планирования, получено в результате оценки работы планировщика PBS (Portable Batch System) промежуточного программного обеспечения gLite, установленного на Grid-кластере Запорожского национального технического университета;

- предполагается, что ВУ (WN\_0\_svc и WN\_1\_svc) характеризуются равной производительностью и решают задачи равной вычислительной сложности. Значение задержки получено в результате оценки времени работы утилиты "PiFast v.1.5" (вычисление числа  $\pi$  с точностью  $10^6$  знаков после запятой) на процессоре AMD с микроархитектурой K10 и частотой 3 ГГц;

- для блока Storage\_svc время задержки выбрано произвольно, т.к. этот блок конечный и на работу системы не влияет;

- значение задержки для блоков Net\_00, Net\_01, Net\_02, Net\_11, Net\_12 задано на основании оценки работы WS "pingtest.net", как усредненное значение времени RTT (Round Trip Time) в пределах европейской зоны.

Таблица 4

Временные задержки на компонентах DEVS-модели

Компонент	Задержка, с
Client_svc	3.00
Planner_svc	0.33
WN_0_svc, WN_1_svc	26.75
Storage_svc	0.01
Net_00, Net_01, Net_02, Net_11, Net_12	0.11

Графическое отображение процесса моделирования системы с CWS в среде DEVS Suite приведено на рис. 2. Здесь показан момент, когда WN\_1\_svc осуществляет обработку заявки job\_0, в то время как Planner\_svc занят обработкой заявки job\_1, осуществляя поиск свободного ВУ (ВУ в состоянии "passive"). На рис. 3. приведено графическое отображение в среде DEVS Suite

фрагмента временных диаграмм работы компонентов Client\_svc и Planner\_svc в процессе моделирования вплоть до данного момента.

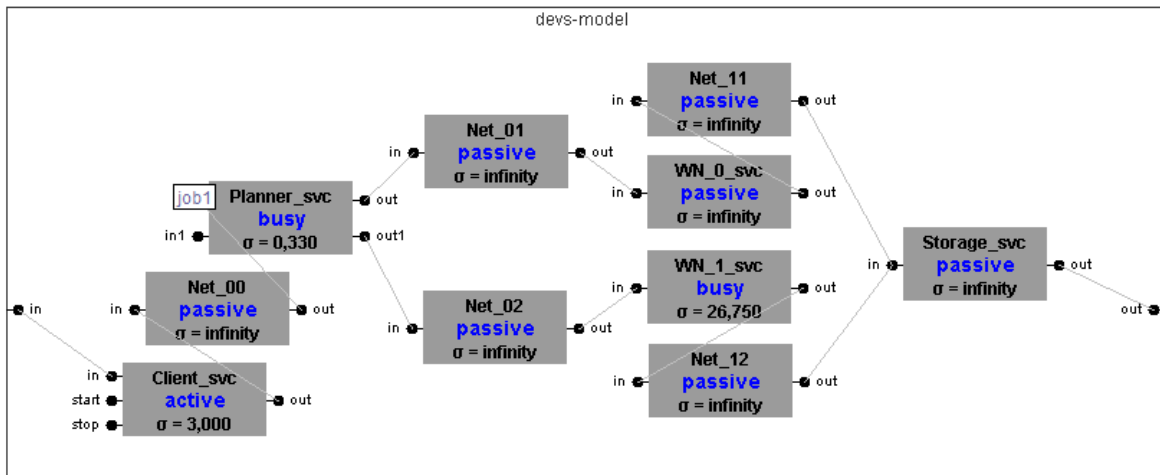


Рис. 2. Графическое отображение структуры системы и процесса моделирования в среде DEVS Suite

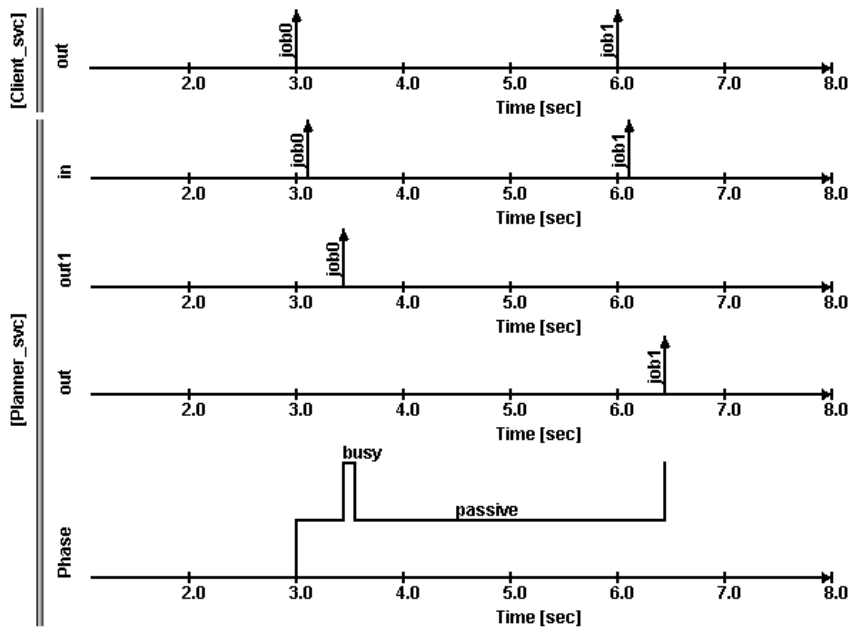


Рис. 3. Фрагмент временных диаграмм процесса моделирования

#### 4. Моделирование системы с CWS в среде GPSS

Рассматриваемую систему с CWS можно представить как модель системы массового обслуживания (СМО) без потерь, состоящую из входного потока заявок  $job_i$ , очереди, обслуживающих приборов и выходящего потока (рис. 4). Компонент Client\_svc является генератором потока  $job_i$ , который передается на компонент Planner\_svc через среду передачи Net\_00. Planner\_svc формирует общую очередь заявок и отправляет заявки через среду передачи Net\_01, Net\_02 к компонентам WN\_0\_svc или WN\_1\_svc, в зависимости от того, который из них находится в состоянии "passive". Далее  $job_i$  поступает в хранилище результатов Storage\_svc через среду передачи данных Net\_11, Net\_12.

Компоненты WN\_0\_svc и WN\_1\_svc образуют двухканальный обслуживающий прибор. Коммуникационные задержки, вносимые средой передачи данных, моделируются одноканальными приборами Net\_00, Net\_01, Net\_02, Net\_11 и Net\_12. Планировщик Planner\_svc моделируется как одноканальный обслуживающий прибор, поскольку, находясь в состоянии "busy", он может обрабатывать только одну заявку  $job_i$ . Компонент Storage\_svc моделируется как уничтожитель поступивших в него заявок, т.к. дальнейшего обслуживания этих заявок производиться не будет.

Значения временных задержек на компонентах модели установлены в соответствии с табл. 4.

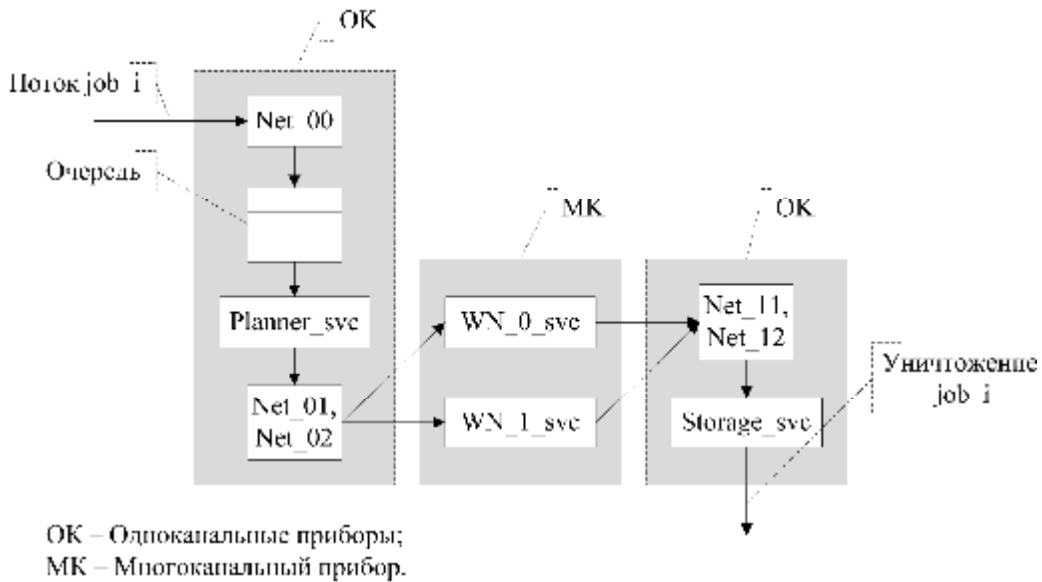


Рис. 4. Структура модели системы с CWS, представленной как СМО

### 5. Анализ результатов моделирования

Моделирование системы с CWS в средах DEVS Suite и GPSS выполнено для различных моделей потоков требований (*job\_i*): равномерного, пуассоновского и нормального.

Математическое ожидание и дисперсия моделей пуассоновского и нормального потоков приведены в табл. 5.

Таблица 5

Параметры потоков	
Поток <i>job_i</i>	Параметры
Пуассоновский	$m=3.00, \sigma^2=3.00$
Нормальный	$m=3.00, \sigma^2=0.36$

В результате многократного повторения моделирования в средах DEVS Suite и GPSS получены усредненные значения следующих характеристик работы СМО:

- общее время вычислительного процесса  $T_w$ ;
- коэффициент загрузки ВУ  $u$ ;
- среднее время нахождения заявок *job\_i* в очереди  $t_q$ .

Таблица 6

Поток <i>job_i</i>	Значения характеристик для DEVS- и GPSS-моделей					
	DEVS-модель			GPSS-модель		
	$T_w, c$	$u$	$t_q, c$	$T_w, c$	$u$	$t_q, c$
Равномерный	675.41	0.993	259.00	675.41	0.993	259.36
Пуассоновский	675.21	0.994	260.22	673.43	0.994	263.07
Нормальный	675.56	0.992	260.32	676.85	0.990	263.56

Из табл. 6 видно, что результаты моделирования в средах DEVS Suite и GPSS отличаются незначительно, что свидетельствует о равных возможностях проводить дискретно-событийное имитационное моделирование. Однако, в отличие от GPSS, среда DEVS Suite позволяет моделировать функции компонентов системы, а DEVS-формализм – специфицировать эти функции. Эти преимущества позволяют сделать вывод о возможности использования DEVS-моделей для валидации CWS.

### Выводы

Таким образом, в работе был проведен сравнительный анализ сред имитационного дискретно-событийного моделирования DEVS Suite, GPSS, AnyLogic, GridSim, MATLAB.

В результате сравнительного анализа в качестве инструментариев моделирования, используемых при построении имитационных моделей систем с CWS, были выбраны среды DEVS Suite и GPSS, причем, практическое моделирование показало, что полученные характеристики CWS отличаются незначительно. Возможности программирования поведения WS и композиции CWS из AWS явились решающими в выборе DEVS Suite в качестве средства валидации CWS.

1. Service oriented architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS). Revised Submission [Електронний ресурс] // OMG document: ad/2008-08-04. – Режим доступа: \www/ URL: <http://www.omg.org/>
2. Cox D.E. Management of the service-oriented architecture life cycle / D.E. Cox, H. Kreger // IBM Systems Journal, 2005. – Vol. 44, No. 4. – P. 709-726.
3. 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.
4. Mittal S. DEVS/SOA: A Cross-Platform Framework for Net-centric Modeling & Simulation in DEVS Unified Process / S. Mittal, J.L. Risco-Martin, B.P. Zeigler // Simulation, 2009. – Vol. 85. – P. 419-450.
5. Томашевський В.М. Моделювання систем / В.М. Томашевський. – К.: Видавнична група BHV, 2005. – 352 с. – ISBN 966-552-120-9.
6. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем / Е.М. Кудрявцев. – М.: ДМК Пресс, 2004. – 320 с. – ISBN 5-94074-219-X.
7. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2005. – 400 с. – ISBN 5-94157-148-8.
8. Buyya R. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing / R. Buyya, M. Murshed // Concurrency and Computation: Practice and Experience, 2002. – Vol. 14, No. 13-15. – P. 1175-1220.
9. Юсупова Н.И. О реализации дискретно-событийных моделей в среде MATLAB / Н.И. Юсупова, О.Н. Сметанина // Управление в сложных системах. – Уфа, 1999. – С. 69-78.
10. Сидоренко В.Н. Имитационное моделирование в науке и бизнесе: подходы, инструменты, применение / В.Н. Сидоренко, А.В. Красносельский // Бизнес-информатика, 2009. – Т. 8, № 2. – С. 52-57.