# БИЗНЕС-ИНФОРМАТИКА

УДК004.414.2.519.876

## МОДЕЛИРОВАНИЕ СЕРВИСНОЙ КОМПОЗИЦИИ С ПОМОЩЬЮ ОКРАШЕННЫХ СЕТЕЙ ПЕТРИ

**И.В. Артамонов**
Байкальский государственный университет экономики и права
E-mail: dark@darkis.ru

Статья описывает свойства сервиса как программной единицы и предлагает рассмотреть аспекты его моделирования с помощью аппарата окрашенных сетей Петри. Для этого сервис представляется как конечное множество операций, обладающих предусловиями и постусловиями вызова. Каждая операция может участвовать в сервисных композициях, а каждая композиция может быть рекурсивно представлена как самостоятельная операция.

*Ключевые слова*: сервис-ориентированная архитектура, сервис, веб-сервис, сервисная композиция, сети Петри, окрашенные сети Петри, WF-сети.

## MODELING OF SERVICE COMPOSITION VIA COLOURED PETRI NETS

**I.V. Artamonov**
Baikal State University of Economics and Law
E-mail: dark@darkis.ru

The article describes service properties as a program unit and suggests considering aspects of its modeling via coloured Petri nets apparatus. For this purpose the service is presented as a finite set of operations with pre- and postconditions for activation. Each operation may participate in service compositions, and each composition may be recursively represented as an independent operation.

*Key words*: service oriented architecture, service, Web service, service compositions, Petri nets, coloured Petri nets, WF-nets.

В настоящее время одной из основных концепций построения распределенных информационных систем различных областей применения считается сервис-ориентированный подход [3–6, 16]. В основе этой концепции, как и в основе сервис-ориентированного программирования лежит понятие сервиса или службы – самостоятельной программной единицы с четко

определенным интерфейсом, который обеспечивает слабую привязку программы к внешней среде.

Выделим следующие свойства сервиса [2]:

• Сервисом может быть как отдельное приложение, так и его модули, процедуры или функции. Приложение может состоять из нескольких сервисов и само, в свою очередь, является службой.

• Службы можно комбинировать и перекомбинировать в сервисные композиции, различные решения и сценарии в соответствии с новыми требованиями предметной области.

• У служб нет состояний. В общем случае службы не зависят от окружающей среды и не меняют своего поведения в зависимости от состояния других служб. Каждый очередной процесс коммуникации со службой не зависит от предыдущих.

• Слабая связанность. Службы выполняют определенную работу по запросам от других служб и возвращают определенный результат. Детали их реализации полностью скрыты, поэтому, как и в случае с компонентами, внутреннюю структуру и реализацию службы можно свободно изменять.

• Сервис связан со средой через интерфейс. Интерфейс описывает все зависимости службы от внешней среды. Он описывает все операции, которые может выполнять служба, все операции, выполнение которых она может затребовать в процессе своей работы, все входные и выходные данные.

• Повторное использование служб. После разработки любой из сервисов можно использовать повторно для решения новых задач. Например, служба, отвечающая за печать определенных документов, может использоваться разными потребителями этой услуги.

• Облегченная интеграция. Благодаря тому, что в основе построения и взаимодействия сервисов лежат открытые технологии [1], нет необходимости в дополнительном программировании средств интеграции.

• Не имеет значение расположение служб. Благодаря ориентации на открытые стандарты, распределенные реестры и интерфейсы, службы могут быть территориально распределены и возможно даже не все подключены к работе в определенный момент времени.

• Динамическое обнаружение служб. Служба может быть обнаружена, подключена и использована приложением сразу во время выполнения.

Одно из направлений развития теории сервис-ориентированной архитектуры и сервис-ориентированного программирования – разработка подходов к моделированию служб и их композиций. Зачастую для этого предлагается использовать аппарат сетей Петри, который за последние 20 лет претерпел серьезное развитие: появилось множество так называемых «расширений», включая направление «высокоуровневых», стохастических, временных и окрашенных сетей. В [7, 11–13, 15, 18, 19] приводятся аргументы, позволяющие считать сети Петри подходящим средством моделирования сервис-ориентированных приложений.

Являясь самостоятельной программной единицей, сервис может реализовывать несколько операций, связанных между собой общностью задач и предметной области. Если рассмотреть некоторые стандарты и подходы к описанию служб (например, стандарт WSDL или технологию Microsoft

WCF [14]), то службу можно представить как совокупность «конечных точек» – сущностей, предоставляющих наборы операций по определенному адресу. Таким образом, в своей совокупности сервис представляет собой множество операций.

Пусть $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ – это окрашенная сеть Петри, моделирующая взаимодействие сервисов [10]. Тогда отдельную операцию службы можно представить как

$$O = (t, P^o, \Sigma^o, V^o, C^o, G^o, E^o),$$

где $t \in T$ – отдельный переход сети Петри;

$P^o \subseteq P$ – множество входных и выходных позиций такое, что $P^o = P_{in} \cup \cup P_{out}$;

$P_{in} \subseteq P^o$ – множество входных позиций перехода $t$;

$P_{out} \subseteq P^o$ – множество выходных позиций перехода $t$;

$A^o \subseteq A = P_{in} \times t \cup t \times P_{out}$ – множество дуг, соединяющих переход $t$ с входными и выходными позициями. Под $A_{in}$ и $A_{out}$ будем понимать набор входящих и исходящих дуг соответственно;

$\Sigma^o \subseteq \Sigma$ – подмножество множества типов данных, используемых в окрашенной сети Петри;

$V^o \subseteq V$ – подмножество типизированных переменных, используемых операцией;

$C^o: P^o \to \Sigma^o$ – функция «окрашивания», ставящая в соответствие каждой позиции подмножество цветов;

$G^o$ – сторожевая функция, ставящая в соответствие операции сторожевое условие, такая, что $Type[G^o(t)] = Bool$, т.е. функция возвращает булевое значение;

$E^o$ – функция выражения над дугой, ставящая в соответствие каждой дуге $a \in A^o$ выражение такое, что $\forall a \in A^o: Type[E^o(a)] = C(p)_{ms}$, где $p$ – это позиция, соединенная с дугой $a$. Под $E_{in}$ и $E_{out}$ будем понимать множество операций над входящими и исходящими дугами соответственно, при этом множество всех выражений над дугами операции можно представить как $E^o = E_{in} \cup E_{out}$.

Единственное сторожевое условие позволяет осуществлять запуск операции в случае, если входные переменные будут удовлетворять условиям, задаваемым стражами. Функция окрашивания $C : P^o \to \Sigma$ позволяет задавать строго типизированный вход и выход операции, что присуще свойствам контракта службы и задачам, которые она реализует.

Предусловия операции (т.е. условия, которые должны быть выполнены перед запуском операции) выражаются с помощью функции окрашивания, задающей определенные типы данных входной позиции и выражений над входными дугами, определяющими наборы входящих данных и/или их преобразования. Т.е. предусловие можно представить как $R_{in} = (P_{in}, A_{in}, E_{in}, \Sigma'_{in}, V'_{in}, C_{in})$. Аналогично представляются постусловия операции (т.е. условия, которые должны быть выполнены при завершении операции): $R_{out} = (P_{out}, A_{out}, E_{out}, \Sigma'_{out}, V'_{out}, C_{out})$.Таким образом, операцию службы можно выразить в терминах пост- и предусловий: $O = (t, R_{in}, R_{out}, G)$.

Ввиду того аппарат окрашенных сетей Петри не позволяет описывать вычисления над данными вне операций над дугами, то выполняемые опе-

рацией функции можно представить только в виде совокупности операций над входными и выходными дугами.

Исходя из определения сети Петри, одна сервисная операция с входами и выходами может представлять собой сеть Петри, являясь простым двудольным ориентированным графом. Пусть $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ – это окрашенная сеть Петри, моделирующая взаимодействие сервисов, а $O = (t, P^o, A', \Sigma', V', C, G, E^o)$ – это отдельная операция сервиса. Композиции операций $o \in O$ являются сетью Петри только тогда, когда они объединены позицией (-ями), входящими во множество позиций каждой операции:

$$\forall o \in O \; \exists \, o' \in O : p \in P^o = p' \in P'^o.$$

При этом операции могут разделять общие позиции. Позиция $p$ операций $O$ и $O'$ является общей между ними в том случае, когда $p \in (P^o \cap P^{o'})$.

Под интерфейсом операции будем понимать совокупность ее входных и выходных позиций, т.е. $P^o$. Назовем две операции $O$ и $O'$ интерфейсно-эквивалентными в том случае, когда $C(\forall p^o \in P^o) = C\,(\forall p^{o'} \in P^{o'})$.

Две операции $O$ и $O'$ назовем интерфейсно-совместимыми в том случае, когда выходные позиции операции $O$ могут являться входными для операции $O'$, т.е. $\forall p \in P^o_{in}, \forall p' \in P^{o'}_{out} : C(p) = C(p')$. Две операции $O$ и $O'$ назовем частично интерфейсно-совместимыми в том случае, когда подмножество выходных позиций операции может входить в подмножество выходных для другой операции, т.е.: $P^o_{in} \cap P^{o''}_{out} \neq \varnothing$. Операции $O$ и $O'$ назовем несовместимыми, если $P^o_{in} \cap P^{o''}_{out} = \varnothing$.

Композицией операций $O$ и $O'$ назовем окрашенную сеть Петри

$$O \oplus O' = (T, P, A, \Sigma, V, C, G, E), \qquad (1)$$

где $T = (t^o, t^{o'})$;

$P = P^o \cup P^{o'}$, причем

$P^o_{in} \cup P^{o'}_{out} \cup P^{o'}_{in} \cup P^o_{out}$, $P_{in} = \left(P^{o \setminus P^{o'}_{out}}_{in}\right) \cup \left(P^{o' \setminus P^o_{out}}_{in}\right)$, $P_{out} = \left(P^{o \setminus P^{o'}_{in}}_{out}\right) \cup \left(P^{o' \setminus P^o_{in}}_{out}\right)$.

Для интерфейсно-совместимых операций в этом случае

$P^o_{in} = P^{o'}_{out} : P_{in} = P^o_{in}, P_{out} = P^{o'}_{out}$;

$A = A^o \cup A^{o'}$;

$\Sigma = \Sigma^o \cup \Sigma^{o'}$;

$C = \{C^o, C^{o'}\}$;

$G = \{G^o, G^{o'}\}$;

$E = \{E^o, E^{o'}\}$.

Композицией конечного множества частично интерфейсно-совместимых операций $O_1 \ldots n$ и $O'$ для всех $n \geq 1$ назовем окрашенную сеть Петри

$$\{O_1 \ldots n\} \oplus O' = (T, P, A, \Sigma, V, C, G, E), \qquad (2)$$

где $T = \{t^{o1} \ldots t^{on}, t^{o'}\}$;

$P = P^{o1} \cup \ldots \cup P^{on} \cup P^{o'}$.

Будем говорить для (2), что имеет место полная совместимость для конечного множества частично интерфейсно-совместимых операций $O_1 \ldots n$ и $O'$ по входу тогда и только тогда, если $P^o_{in} \not\subseteq P_{in}$, и по выходу, если $P^o_{out} \not\subseteq P_{out}$.

Сервис можно представить как множество операций $O$, объединенных общностью бизнес-логики программы. Сервис может реализовывать совершенно различные функции, поэтому к множеству операций невозможно применить требования общности пост- или предусловий. При этом разные операции одного сервиса могут входить в разные сервисные композиции (моделируемые разными сетями Петри). Верно и другое: разные операции сервиса могут входить в одну модель.

Исходя из природы сервиса, каждая операция может быть рекурсивно декомпозирована на ряд вложенных функций до достижения уровня элементарных операций. Рассмотрим операцию $O$ как вложенную окрашенную сеть Петри $CPN'$, содержащую непустое множество переходов. Все предусловия операции $O$ необходимо представить элементами, задающими начальную маркировку $CPN'$, а постусловия задают финальную маркировку.

Пусть $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ – это окрашенная сеть Петри, моделирующая взаимодействие сервисов, а $O = (t, P_o, A', \Sigma', V', C, G, E_o)$ – это отдельная операция сервиса.

Тогда для $O$ будем называть сеть $CPN' = (P_s, T_s, A_s, \Sigma_s, V_s, C_s, G_s, E_s, I_s)$ декомпозирующей сетью 1-го порядка в случае, если:

1. $\forall p_{in} \in P_{in} \exists M_o(p_s) : M_o(p_s) = M(p_{in})$ и $C(p_{in}) = C(p'_{in})$, где $p_s \in P_s$, а для всех остальных $p_s \in P_s$ выполняется условие $M_o(p_s) = \varnothing$.

2. $\forall p_{out} \in P_{out} \exists M_f(p_s) : M_f(p_s) = M(p_{out})$ и $C(p_{in}) = C(p'_{in})$, где $p_s \in P_s$, под $M_f$ понимается финальная маркировка.

3. Существует конечная последовательность шагов, таких что $M_o(p_s) \rightarrow \rightarrow M_f(p_s)$, т.е. финальная маркировка достижима из начальной маркировки $M_f \in R(M_o)$ и $R(M_f) = \varnothing$, т.е. финальная маркировка мертва.

4. $\forall t \in T_s : \exists Y(t) : M'^{Y} \rightarrow ... \rightarrow M_f$, где $(M', M_f) \in R(M_o)$. Это условие позволяет наложить на операцию, представленную окрашенной сетью, условие отсутствия циклов и сегментов сети, не связанных с достижением целевой маркировки.

Таким образом, все входные позиции операции отображаются как начальная маркировка для вложенной сети, при этом остальные позиции не имеют маркировки. Все выходные позиции формируются из финальной маркировки вложенной сети, достижимой из начальной.

Будем называть сеть $CPN^k$ декомпозирующей сетью $k$-го порядка в случае, если она декомпозирует операцию $O$, входящую в сеть $(k-1)$-го порядка. Будем называть операцию атомарной в случае, если она реализуется операцией, и композитной, если она реализуется декомпозирующей сетью.

Отметим, что сеть Петри, реализующая операцию, зависит от внешней среды только через интерфейс операции $P_o$, что позволяет реализовать концепцию слабой связанности. Определение операции в виде низкоуровневой сети Петри позволяет реализовать рекурсивную декомпозицию этой сети на операции (сети) более низкого уровня. Верно и обратное: сеть Петри, удовлетворяющую условиям сервисной операции, можно представить как реализацию некоторой высокоуровневой функции.

Таким образом, представленная методика выделения из службы набора операций позволяет реализовать модели сервисного взаимодействия на сетях Петри без ограничений, связанных со сложностью службы. В отличие от представленных в западной литературе подходов в виде «сервисных» или «открытых» сетей Петера Массутхе (англ. Peter Massuthe) (например, в [12]), workflow-сетей Виллаван дер Аалста (англ. Will.M.P. vanderAalst) [17] и простых сетей Петри, представленная модель обладает следующими преимуществами.

– Предлагает рассматривать сервис и с точки зрения программной реализации, т.е. позволяет описывать не сервисы целиком, а отдельные сервисные операции, которые, возможно, не связаны друг с другом.

– Выделяет ключевые аспекты сервис-ориентированного программирования: понятие интерфейса, независимой реализации службы, сервисной композиции, которая может рассматриваться как единый сервис с более высокого уровня абстракции и также включать в композицию с другими службами.

– Описывается на базе высокоуровневых сетей Петри – окрашенных сетей, что позволяет использовать все преимущества этой технологии для моделирования динамических систем (они были описаны, например, в [8–10]), в первую очередь, получаемая модель, в отличие от простых сетей Петри, способна описывать манипуляции с различными типами данных.

– Не имеет ограничений, накладываемых на процесс выполнения на начальные и финальные позиции модели, что позволяет считать взаимодействие завершенным при достижении финальной маркировки, а не некоторой финальной позиции. Верно и обратное – инициировать взаимодействие позволяет начальная маркировка, охватывающая, возможно, множество сервисов, а не некоторая начальная позиция.

**Литература**


1. *Артамонов И.В.* Описание бизнес-процессов: вопросы стандартизации // Прикладная информатика. 2011. № 3. С. 20–28.
2. *Артамонов И.В.* Разработка распределенных сервисно-ориентированных программных средств. Иркутск: БГУЭП, 2012. 130 с.
3. *Bieberstein Norbert et al.* Executing SOA: A Practical Guide for the Service-Oriented Architect. IBM Press, 2008. P. 240.
4. *Brown Paul C.* Implementing SOA: Total Architecture in Practice. Addison Wesley Professional, 2008. P. 736.
5. *Catts Anthony, St.Clair Joseph.* Business Process Managment enabled by SOA. IBM Press, 2009. P. 82.
6. *Cohen Frank.* FastSOA. Elsevier Inc, 2007. P. 278.
7. *Graham Ian.* Requirements Modelling and Specification for Service Oriented Architecture. John Wiley & Sons Ltd, 2008. P. 301.
8. *Jensen Kurt.* An Introduction to the Theoretical Aspects of Coloured Petri Nets // A Decade of Concurrency, Lecture Notes in Computer Science / de Bakker J.W., de Roever W.-P., Rozenberg G. Springer-Verlag, 1994. Vol. 803.
9. *Jensen Kurt, Kristensen Lars Michael, Wells Lisa.* Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems // International Journal on Software Tools for Technology Transfer (STTT). 2007. Vol. 9. P. 213–254.
10. *Jensen Kurt.* Coloured Petri Nets modeling and validation of concurrent systems. Springer, 2009. P. 384.

11. *Kohler Michael, RolkeHeiko*. Web Service Orchestration with Super-Dual Object Nets // Petri Nets and Other Models of Concurrency / Kleijn Jetty, Yakovlev Alex. Springer, 2007.
12. *Massuthe Peter*. Operating Guidelines for Services: Dissertation. University Press Facilities, 2009. P. 266.
13. *Men P., Duan Z., Yu B*. Utilizing Fuzzy Petri Net for Choreography Based Semantic Web Services Discovery // Petri Nets and Other Models of Concurrency / Kleijn Jetty, Yakovlev Alex. Springer, 2007.
14. *Pathak Nishith*. Pro WCF 4.0: Practical Microsoft SOA Implementation. Apress, 2011. P. 446.
15. *Popescu Corina, Soto M. Cavia, Lastraa Jose L. Martinez*. A Petri net-based approach to incremental modelling of flow and resources in service-oriented manufacturing systems // International Journal of Production Research. 2012. Vol. 50. P. 325–343.
16. *Rosen Mike et al*. Applied SOA: Service-Oriented Architecture and Design Strategies. Wiley Publishing, Inc., 2008. P. 662.
17. *Van der Aalst W.M.P*. The Application of Petri Nets to Workflow Management // Journal of Circuits, Systems and Computers. 1998. Vol. 8. P. 21–66.
18. *Wolf Karsten*. Does My Service Have Partners? // Transactions on Petri Nets and Other Models of Concurrency II / Jensen Kurt, van der Aalst Wil M.P. Springer, 2009.
19. *Zafar Bassam*. Conceptual Modelling of Adaptive Web Services based on High-level Petri Nets: PhD Thesis. De Montfort University, 2008. P. 188.

**Bibliography**

1. *Artamonov I.V*. Opisanie biznes-processov: voprosy standartizacii // Prikladnaja informatika. 2011. № 3. P. 20–28.
2. *Artamonov I.V.* Razrabotka raspredelennyh servisno-orientirovannyh programmnyh sredstv. Irkutsk: BGUJeP, 2012. 130 p.
3. *Bieberstein Norbert et al*. Executing SOA: A Practical Guide for the Service-Oriented Architect. IBM Press, 2008. P. 240.
4. *Brown Paul C*. Implementing SOA: Total Architecture in Practice. Addison Wesley Professional, 2008. P. 736.
5. *Catts Anthony, St.Clair Joseph*. Business Process Managment enabled by SOA. IBM Press, 2009. P. 82.
6. *Cohen Frank*. FastSOA. Elsevier Inc, 2007. P. 278.
7. *Graham Ian*. Requirements Modelling and Specification for Service Oriented Architecture. John Wiley & Sons Ltd, 2008. P. 301.
8. *Jensen Kurt*. An Introduction to the Theoretical Aspects of Coloured Petri Nets // A Decade of Concurrency, Lecture Notes in Computer Science / de Bakker J.W., de Roever W.-P., Rozenberg G. Springer-Verlag, 1994. Vol. 803.
9. *Jensen Kurt, Kristensen Lars Michael, Wells Lisa*. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems // International Journal on Software Tools for Technology Transfer (STTT). 2007 г. Vol. 9. P. 213–254.
10. *Jensen Kurt*. Coloured Petri Nets modeling and validation of concurrent systems. Springer, 2009. P. 384.
11. *Kohler Michael, RolkeHeiko*. Web Service Orchestration with Super-Dual Object Nets // Petri Nets and Other Models of Concurrency / Kleijn Jetty, Yakovlev Alex. Springer, 2007.
12. *Massuthe Peter*. Operating Guidelines for Services: Dissertation. University Press Facilities, 2009. P. 266.
13. *Men P., Duan Z., Yu B*. Utilizing Fuzzy Petri Net for Choreography Based Semantic Web Services Discovery // Petri Nets and Other Models of Concurrency / Kleijn Jetty, Yakovlev Alex. Springer, 2007.
14. *Pathak Nishith*. Pro WCF 4.0: Practical Microsoft SOA Implementation. Apress, 2011. P. 446.

15. *Popescu Corina, Soto M. Cavia, Lastraa Jose L. Martinez*. A Petri net-based approach to incremental modelling of flow and resources in service-oriented manufacturing systems // International Journal of Production Research. 2012. Vol. 50. P. 325–343.
16. *Rosen Mike et al*. Applied SOA: Service-Oriented Architecture and Design Strategies. Wiley Publishing, Inc., 2008. P. 662.
17. *Van der Aalst W.M.P*. The Application of Petri Nets to Workflow Management // Journal of Circuits, Systems and Computers. 1998. Vol. 8. P. 21–66.
18. *Wolf Karsten*. Does My Service Have Partners? // Transactions on Petri Nets and Other Models of Concurrency II / Jensen Kurt, van der Aalst Wil M.P. Springer, 2009.
19. *Zafar Bassam*. Conceptual Modelling of Adaptive Web Services based on High-level Petri Nets: PhD Thesis. De Montfort University, 2008. P. 188.