

DOI: 10.17117/na.2018.04.03.030

Поступила (Received): 03.04.2018

<http://ucom.ru/doc/na.2018.04.03.030.pdf>

Гагиева З.О., Иванова И.В.
Моделирование web-сервиса с помощью сети Петри

Gagieva Z.O., Ivanova I.V.
Web service modelling with Petri Nets

В этой статье исследован пример моделирования web-сервиса с помощью сети Петри. Разработана модель сети Петри для web-сервиса. Так же подробно описываются этапы работы сервиса и сопоставляются с моделью СП. Web-сервис был разработан на языке программирования PHP, в среде разработки PhpStorm. Для моделирования и анализа пространства состояний применяется специализированный редактор сетей Петри

Ключевые слова: сеть Петри, web-сервис, редактор сети Петри, PhpStorm

In this article, we studied an example of a web service modelling using Petri Nets (PN). The model of Petri Net for web-service was developed. The stages of the service operation are described in detail and are compared with the PN model. The Web service was developed with PHP programming language, in the PhpStorm development environment. A specialized Petri Net editor is used for modelling and analysis of the state space

Key words: Petri Net, web service, Petri Net editor, PhpStorm

Гагиева Зарина Олеговна

Бакалавр

Санкт-Петербургский горный университет

г. Санкт-Петербург, 21 линия В.О., 1

Gagieva Zarina Olegovna

Bachelor

Saint Petersburg mining university

St. Petersburg, 21 Liniya V.O., 1

Иванова Ирина Владимировна

Доктор технических наук, профессор, почетный работник высшего профессионального образования РФ

Санкт-Петербургский горный университет

г. Санкт-Петербург, 21 линия В.О., 1

Ivanova Irina Vladimirovna

Doctor of Technical Sciences, Professor, Honorary Worker of higher professional education

Saint Petersburg mining university

St. Petersburg, 21 Liniya V.O., 1

В настоящее время трудно представить успешную, высокопроизводительную работу предприятий без качественных программных продуктов. Для минимизации вкладываемых ресурсов в разработку программных продуктов пользуются специальными методами анализа, синтеза, проектирования и реализации. Сети Петри (СП) являются одним из наиболее популярных формализмов для моделирования и анализа параллельных и распределенных систем [1]. Анализ сетей Петри помогает получить важную информацию о структуре и динамическом поведении моделируемой системы. Эта информация может быть полезна для оценки моделируемой системы и выработки предложений по её усовершенствованию и изменению [2]. Анализ результатов может сказать о том, какие действия происходят, какие состояния им предшествовали, и какие состояния примет система после выполнения действия. Сеть Петри моделирует некоторую структуру и динамику ее функционирования [3].

Моделирование web-сервиса. Этапы разработки web-сервиса

Для того чтобы четко поставить задачу на разработку web-сервиса, желательно предварительное моделирование как структуры системы, так и процессов, происходящих в ней.

Рассмотрим этапы разработки web-сервиса на примере auto web сервиса. Рассматриваемый web-сервис предназначен для получения, либо хранения полезной информации. Создан класс, в котором хранится информация (например, марки автомобилей, названия растений, номера мобильных телефонов, пароли от аккаунтов и т. п.); каждой марке присвоен номер. Для получения информации (в нашем случае названия марки автомобиля) пользователю нужно обратиться к web-сервису с запросом, в котором будет находиться номер искомой марки, после чего сервис выдаст на экран название марки, соответствующее этому номеру.

Разработанная сеть Петри представлена на рисунке 1.

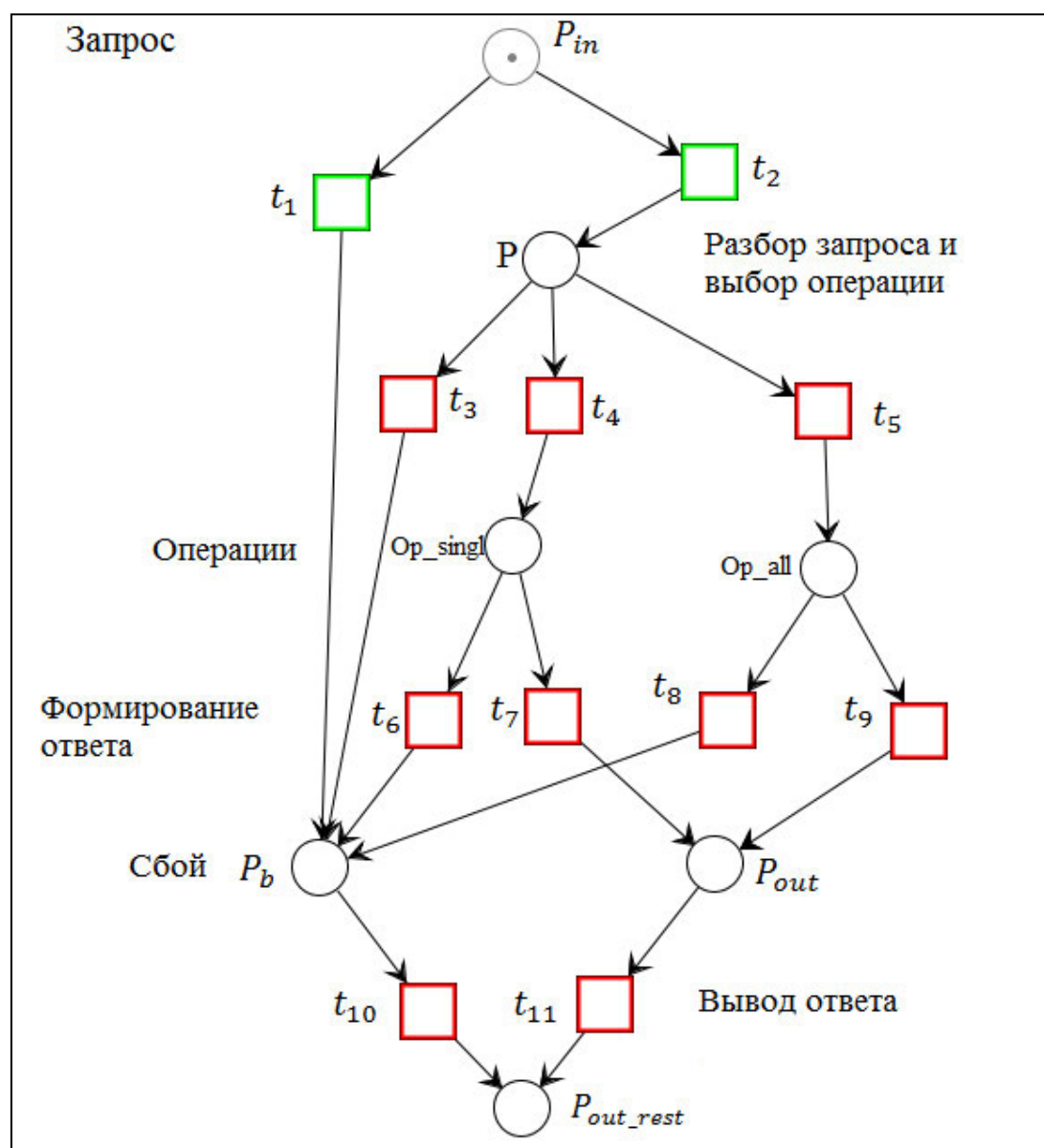


Рис. 1. Модель web-сервиса в виде сети Петри

Модель, показанная на рис. 1 была выполнена в специализированном редакторе для сетей Петри. Зеленый цвет означает, что переход возбужден, и метка может перейти в позицию P_b , либо P . Красный цвет означает, что переход сработать не может. Переход срабатывает, если в каждой входной позиции (предусловии) число меток не меньше числа дуг, исходящих из позиции в данный переход. Такие переходы называют возбужденными, их срабатывание может наступить через любой конечный промежуток времени после возбуждения. В результате срабатывания из всех входных позиций перехода исключается число меток равное числу дуг, выходящих из соответствующей позиции в переход, а в выходные позиции данного перехода добавляется число меток равное числу дуг исходящих из перехода в соответствующую выходную позицию.

Модель, изображенная на рис. 1 состоит из 6 основных процессов, которые описывают основные этапы работы web-сервиса. Опишем код программы. Покажем элементы, которые отвечают за процессы работы web-сервиса.

1) Запрос отправляется на сервер в виде адреса web-сервиса к которому обращается пользователь. Если запрос некорректный, то на экране пользователя отобразиться ошибка. Если же сервер находит по адресу web-сервис, то следующим шагом будет разбор запроса и выбор операции.

В сети Петри (рис.1) запрос это метка, находящаяся в позиции P_{in} . При попадании метки в позицию P_{in} возбуждаются два перехода t_1 и t_2 , то есть может сработать любой из них.

Далее показан фрагмент кода (PHP), отвечающий за рассмотренную операцию.

Разбор запроса:

```
$view="";
if (isset($_GET["view"]))
$view = $_GET["view"];
```

2) Следующий этап это – выбор операции. В модели сети Петри (рис. 1) выбор операции происходит после срабатывания перехода t_2 , и попадания метки (метку также можно назвать фишкой) в позицию P . В зависимости от того что пользователь запрашивает, выбирается операция.

– Если пользователь запрашивает определенную марку автомобиля, то срабатывает «case “single”», *обработка REST (архитектура web-сервиса) адресов /auto/list/<id>/*.

В СП срабатывает переход t_4 и метка переходит в позицию Op_singl (рис. 1).

– Если пользователь запрашивает все марки автомобилей, то срабатывает «case “all”», *обработка REST адресов /auto/list/*. В СП срабатывает переход t_5 и метка помещается в позицию Op_all (рис. 1).

– Если запрос пользователя пуст, то он получает ошибку. В СП срабатывает переход t_3 и метка переходит в позицию P_b (сбой).

Фрагмент PHP кода. Выбор операции:

```
switch ($view) {
case "all":
// Обработка REST адресов /auto/list
```

```

$autoRestHandler = new AutoRestHandler();
$autoRestHandler->getAllAutos();
break;
case "single":
// Обработка REST адресов /auto/list/<id>/
$autoRestHandler = new AutoRestHandler();
$autoRestHandler->getAuto($_GET["id"]);
break;
case "":
// 404 – страница не найдена
break;
}

```

3) Операции. После выбора операции web-сервис приступает к самим операциям. Рассмотрены два случая:

1. Ниже показан фрагмент кода «операция выбора всех автомобилей», эта часть web-сервиса проверяет, имеются ли в базе (класс в котором хранится информация) запрашиваемые автомобили или нет.

В СП происходит выбор перехода из позиции Op_all: если автомобили не найдены, срабатывает переход t_8 и метка переходит в позицию P_b (сбой). Иначе, срабатывает переход t_9 , и метка помещается в позицию P_{out} .

Фрагмент кода (класс в котором хранится информация):

```

1 => 'Audi R8',
2 => 'Maserati Quattroporte',
3 => 'Ferrari 458',
4 => 'Lamborghini Gallardo',
5 => 'Nissan GT-R',
6 => 'Mercedes AMG S 63',
7 => 'BMW M6');

```

Фрагмент PHP кода. Операция выбора всех автомобилей:

```

function getAllAutos()
{
    $auto = new Auto();
    $rawData = $auto->getAllAutos();
    if (empty($rawData)) {
        $statusCode = 404;
        $rawData = array('error' => 'Автомобили не найдены!');
    } else {
        $statusCode = 200;
    }
}

```

2. Во втором случае происходит аналогичная ситуация, только часть web-сервиса проверяет наличие одного определенного автомобиля, а не всех.

В СП происходит выбор перехода из позиции Op_singl: если автомобили не найдены, срабатывает переход t_6 и метка переходит в позицию P_b (сбой). Иначе, срабатывает переход t_7 , и метка помещается в позицию P_{out} .

Фрагмент кода. Операция выбора автомобиля с указанным id:

```
public function getAuto($id)
{
    $auto = new Auto();
    $rawData = $auto->getAuto($id);
    if (empty($rawData)) {
        $statusCode = 404;
        $rawData = array('error' => 'Автомобиль не найден!');
    } else {
        $statusCode = 200;
    }
}
```

4) Формирование ответа. В СП этот процесс происходит при попадании метки в позицию P_{out} .

Функция для формирования ответа:

```
public function prepareResponse($responseData)
{
    $htmlResponse = "<table border='1'>";
    foreach ($responseData as $key => $value) {
        $htmlResponse .= "<tr><td>". $key . "</td><td>". $value . "</td></tr>";
    }
    $htmlResponse .= "</table>";
    return $htmlResponse;
}
```

5) Следующий шаг web-сервиса это сохранение ответа в переменную.

Фрагмент кода:

```
$response = $this->prepareResponse($rawData);
```

6) Вывод ответа. В СП это срабатывание перехода t_{10} или t_{11} . Для вывода ответа web-сервиса создается функция для отображения ответа.

Функция для отображения ответа:

```
public function showResponse($response)
{
    // echo – команда PHP для вывода информации
    echo $response;
}
```

Вывод. Представленная модель web-сервиса в виде сети Петри иллюстрирует возможность появления конфликтных ситуаций. С помощью этой модели разработан web-сервис, в котором учитываются возможные конфликтные ситуации. Web-сервис был разработан на языке программирования PHP, в среде разработки PhpStorm. Для моделирования и анализа пространства состояний, использовали специализированный редактор сетей Петри.

Список используемых источников:

1. Башкин В.А. Некоторые методы ресурсного анализа сетей Петри. Диссертация на соискание степени доктора физ-мат наук. Ярославль. 2014. 44 с.
URL: http://нэб.рф/catalog/000199_000009_005547251/viewer/?page=37

2. Советов Б.Я. Моделирование систем. М.: Высш. шк., 2001. 343 с.
3. Мальков М.В., Малыгина С.Н. Сети Петри и моделирование // Труды Кольского научного центра РАН. 2010. №3. С. 35-40.
4. Аунг Аунг Хейн, Щукин Б.А. Веб-сервис для моделирования процесса инкассации // Современные технологии в задачах управления, автоматизации и обработки информации. 2010. Алушта. С. 251-252.

© 2018, Гагиева З.О., Иванова И.В.

Моделирование web-сервиса с помощью сети Петри

© 2018, Gagieva Z.O., Ivanova I.V.

Web service modelling with Petri Nets