

3. Орлов А.И. Методы принятия управленческих решений (для бакалавров). — М. : КноРус, 2018. — 317 с.
4. Семенов С.С., Воронов Е.М., Полтавский А.В. Методы принятия решений в задачах оценки качества и технического уровня сложных технических систем. — М. : Ленанд, 2016. — 520 с.
5. Стинбарджер Б. Психология трейдинга. Инструменты и методы принятия решений. — М. : Альпина Паблишер, 2017. — 364 с.

УДК 519.68

АВТОМАТИЗИРОВАННОЕ МОДЕЛЬНОЕ ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ

Д.Б. Сабиров, О.М. Гущина

Тольяттинский государственный университет, г. Тольятти

Аннотация. Информационная система (ИС) — это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи экономической информации и принятия управленческих решений. От нее зависит эффективная работа всего предприятия, использующего ИС. Важным качеством информационной системы является ее надежность. Для поддержания корректной работы информационной системы требуется производить контроль и тестирование ее работы. В данной статье приводится теоретическая информация о специальной тестирующей системе, которая будет контролировать и автоматически тестировать различные процессы, происходящие в ИС. Данная тестирующая система основана на теории сетей Петри. Главным требованием к тестирующей системе является ее гибкость. Так как информационные системы быстро развиваются и постоянно изменяются, требуется, чтобы внесения изменений в тестирующую систему производились легко. Благодаря наглядному представлению сетей Петри работать с ними достаточно просто. Основная идея заключается в том, что создается модель всей информационной системы или конкретной ее части, которая нуждается в тестировании. Затем к этой сети Петри с использованием интеграционных модулей пишутся тесты, а после отправляются в самую тестирующую систему, которая будет взаимодействовать с реальной информационной системой и будет проверять ее соответствие мо-

дели. Тестовые сценарии будут генерироваться автоматически. Таким образом, нет необходимости их продумывать, что значительно сокращает время на написание тестов. Благодаря динамике сетей Петри удастся находить совершенно неожиданные некорректные состояния системы.

Ключевые слова: информационные системы, модельное тестирование, сети Петри

AUTOMATED MODEL-BASED TESTING OF INFORMATION SYSTEMS USING PETRI NETS

D.B. Sabirov, O.M. Gushchina
Togliatti State University, Togliatti

Abstract. Information system (IS) are formal, sociotechnical, organizational systems designed to collect, process, store, and distribute information. In a sociotechnical perspective, information systems are composed by four components: task, people, structure (or roles), and technology. The effective work of the entire enterprise depends on it. An important quality of an information system is its reliability. To maintain the correct operation of the information system, it is required to monitor and test its operation. This article provides theoretical information about a special testing system that will monitor and automatically test various processes that occur in IS. This testing system is based on the theory of Petri nets. The main requirement for a testing system is its flexibility. Since information systems are rapidly evolving and constantly changing, it is required that changes to the testing system be made easily. Due to the visual representation of Petri nets, working with them is quite simple. The main idea is to create a model of the entire information system or a specific part of it that needs to be tested. Then, tests are written to this Petri net using integration modules, and then they are poisoned into the testing system itself, which will interact with the real information system and check its conformity to the model. Test cases will be generated automatically. Thus, there is no need to think through them, which significantly reduces the time for writing tests. Thanks to the dynamics of Petri nets, it is possible to find completely unexpected incorrect states of the system.

Keyword: information system, model-based testing, Petri net

Информационная система (ИС) — это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала,

предназначенная для сбора, обработки, хранения и выдачи экономической информации и принятия управленческих решений.

Как видно из определения информационной системы, она отвечает за множество совершенно различных процессов управления, хранения, обработки и передачи информации. Все эти сложные процессы можно описать любой системой условных обозначений для моделирования бизнес-процессов (IDEF0, BPMN, FlowChart, eEPC) [5]. Главным недостатком всех перечисленных нотаций является то, что они не имеют под собой четкой математической базы. Все эти нотации хоть и описывают поведение системы, но не могут быть использованы для программной проверки соответствия реальной системы. Чтобы решить проблемы, описанные выше, в тестирующей системе для описания модели используются сети Петри [6].

Сети Петри — математический аппарат для моделирования динамических дискретных систем [1]. Сеть Петри можно представить в виде двудольного графа, где вершины бывают двух типов — позиции и переходы. Дуги могут соединять только вершины разного типа. В каждой позиции могут размещаться метки, которые также могут перемещаться от вершины к вершине по дугам. Переход срабатывает при проходе через него метки. Сети Петри имеют четкое математическое описание, следовательно, модель системы можно анализировать. Также существуют подходы, которые позволяют преобразовывать BPMN-модель в модель, описанную сетью Петри [3]. Такая возможность позволяет с небольшими затратами перейти на тестирование с помощью сетей Петри.

Тестирующая система состоит из нескольких модулей (рис. 1).

1. Среда разработки сети Петри — это среда, которая позволяет спроектировать сеть Петри и выдать ее описание в одном из поддерживаемых тестирующей системой форматов.

2. Модуль симуляции — это основной модуль тестирующей системы: в нем выполняется моделирование поведения сети Петри.

3. Модуль тестирования — связывает модуль симуляции с тестируемой системой.

Рассмотрим теперь более подробно каждый модуль.

В роли среды проектирования может выступать любая программа, способная выводить сеть Петри в формате .cnp. Также допусти-

мо использовать любой другой формат представления сети Петри. Для этого лишь требуется написать специальный конвертер под интерфейс тестирующей системы.

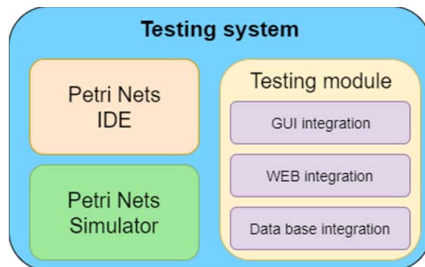


Рис. 1. Схема тестирующей системы

Модуль симуляции принимает на вход XML-файл формата .crn, который содержит описание сети Петри. При запуске модуля симуляции он будет моделировать поведение сети Петри и отправлять управляющие команды модулю тестирования.

Модуль тестирования содержит в себе различные модули интеграции с системой. С помощью этих модулей проводится проверка системы, то есть выполнения некоторых действий в ней, и оценка соответствия ожидаемого и получившегося состояния системы. Весь код, который описывает тесты, записывается в модуле тестирования. Также модуль тестирования должен определять момент завершения тестирования и формировать отчет о неправильном поведении системы и вести журнал исполнения.

Благодаря такой схеме проектирования система тестирования остается очень гибкой с возможностью заменять модули под необходимые конфигурации. Все модули разделены, и нет никакой привязки к конкретным системам или языкам программирования. Модуль симуляции абсолютно независим. Он принимает от конверторов сеть Петри, а затем выполняет симуляцию ее поведения. Модуль симуляции только отправляет команды о том, в какое состояние перешла система и какое действие необходимо выполнить. Этому модулю совершенно не важно, какой процесс он симулирует. Это может быть некоторая обработка данных, их передача или просто взаимодействие сотрудников. Всю связующую работу выполняет модуль тестирова-

ния. Он является некоторым адаптером, преобразующим команды от модуля симуляции к объекту тестирования и наоборот. Также модуль симуляции имеет различные настройки своего поведения, которые можно настроить для лучшей покрываемости.

Опишем, как строится сеть Петри. В роли вершины будет выступать состояние некой сущности в системе. Это может быть открытое окно у пользователя на экране, этап обработки документа, состояние некоторого бизнес-процесса. В роли перехода будет выступать некоторое действие в системе. Когда выполняется переход, система меняет свое состояние. Так, допустим, нажатие некоторой клавиши, одобрение заявки, отправка сообщения могут выступать в роли перехода. Меткой является сам объект в системе, например пользователь, документ и т. д.

После того как описана модель тестируемой системы, необходимо для каждого состояния и каждого перехода написать код тестирования. В вершинах мы выполняем проверку корректности системы. Важно, чтобы код проверки не изменял состояние системы, а выполнял лишь проверку ее корректности. Код, описывающий переход, должен выполнять некоторое действие в системе. Проверку самой тестируемой системы производить не нужно.

Таким образом, тестирование выполняется за счет того, что происходит симуляция работы сети Петри. Метки, проходя через переходы и меняя состояния, вызывают код тестирования для этих состояний и переходов. Когда метка проходит через переход, она активирует код выполнения некоторого действия в системе, а затем, когда метка попадает в вершину, выполняется проверка, в корректном ли состоянии оказалась система после этого перехода.

В сети Петри может возникать ситуация, когда могут активироваться сразу два перехода (рис. 2).

Так как переходы активируются равновероятно, то могут возникать ситуации, что при малом количестве тестов один переход активируется чаще другого [2].

Для решения этой проблемы на переходы добавляется некоторая вероятность их активации.

В случае, когда переход активируется, его вероятность активации снижается. Такой подход значительно увеличивает покрываемость.

мость сети Петри различными путями, что позволяет генерировать более разнообразные тестовые сценарии.

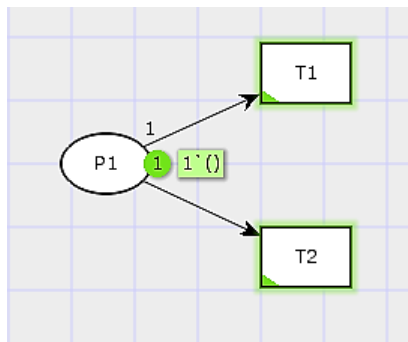


Рис. 2. Ситуация, в которой могут быть активированы оба перехода

Еще одной особенностью является то, что сеть Петри может содержать циклы, следовательно, количество различных путей в ней не ограничено.

Чтобы тестирование не выполнялось бесконечно, необходимо добавить ограничения на выполнение. В качестве одного из решений предлагается прекращать тестирование после прохода по всем ребрам или посещения всех состояний. Использование такого подхода сильно зависит от самой сети [4; 7]. Лучшим вариантом предлагается использование ограничения по количеству итераций в сети Петри.

Таким образом, появляется некоторая величина, индивидуальная для каждой системы, которая дает оптимальную покрываемость системы тестовыми сценариями. Также это позволяет контролировать общее время тестирования.

В информационной системе присутствуют объекты различного типа, такие как документы, сотрудники, задачи и т. д. Каждый объект содержит определенный набор атрибутов, которые влияют на то, как этот объект взаимодействует с системой. Допустим, каждый пользователь информационной системы имеет свой набор прав и ограничений на выполнение определенных действий. Такие атрибуты необходимо учитывать, и важно, чтобы тестирующая система давала такую возможность. Именно поэтому в модуль симуляции сети Петри включена поддержка цветных сетей Петри.

Цветная сеть Петри — это сеть Петри, в которой метки имеют различный тип. Этот тип может быть очень сложным. Каждая позиция в такой сети также имеет свой цвет, и хранить она может только фишки такого же цвета, что и она сама.

Таким образом, на исходящие ребра добавляется условие, которое будет проверять тип и наличие некоторого атрибута. Благодаря использованию цветных сетей Петри диаграмма этой сети становится меньше и более читаемой.

Предложенная тестирующая система помогает определить цвет вершины еще на этапе построения сети Петри. Такой подход позволяет при написании тестов работать сразу с объектом нужного типа без необходимости дополнительно сверяться с документацией. Модуль симуляции будет автоматически сверять типы и дополнительно проверять допустимость срабатывания перехода, обращаясь к одному из модулей интеграции за информацией о наличии необходимого атрибута.

Приведем простой пример того, как строится сеть Петри для случая входа пользователя в систему.

Допустим, у нас имеется некоторое окно входа в систему, которое требует данные для аутентификации.

В случае если пользователь вводит верные данные, он попадает на главное окно системы, если нет — получает сообщение об ошибке.

В таком случае сеть Петри будет состоять из трех состояний и четырех переходов (рис. 3).

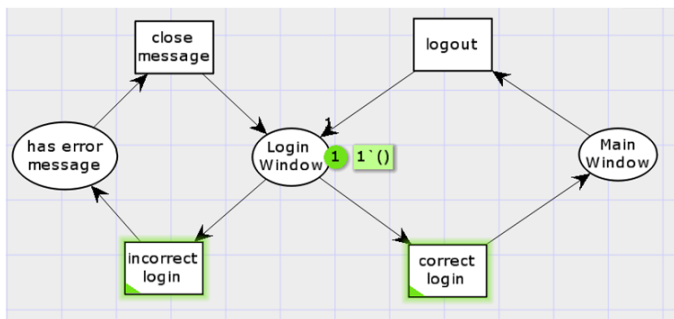


Рис. 3. Сеть Петри для тестирования аутентификации пользователя

В состоянии Login Window необходимо написать код, который выполняет проверку того, что отображено именно окно приветствия и что на нем присутствуют все необходимые элементы.

Далее мы описываем два перехода correct login и incorrect login, в них выполняется ввод правильных и неправильных данных аутентификации.

В зависимости от активировавшегося перехода пользователь либо перейдет в состояние has error message, где будет происходить проверка того, что пользователь не смог войти в систему и получил сообщение об ошибке, либо произойдет переход в состояние Main Window, где будет проверяться то, что пользователю открылось рабочее окно системы, и он вошел, и ему присвоен корректный пользователь системы.

При запуске тестирования этой функциональности будет происходить постоянная симуляция работы сети Петри, которая в свою очередь будет взаимодействовать с системой и проверять правильность ее работы и соответствие составленной модели.

Литература

1. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. — М. : Научный мир, 2004. — 208 с.
2. Орел А.А. Шаблоны проектирования бизнес-процессов с помощью сетей Петри // Математика. Механика. — 2009. — № 11. — С. 45–48.
3. Дорпер М.Г. Алгоритм преобразования моделей бизнес-процессов в одноцветные сети Петри // Моделирование и анализ информационных систем. — 2010. — № 2. — С. 5–16.
4. Hagge N., Wagner B. Java code patterns for Petri net based behavioral models. Industrial Informatics. — 2005. — P. 450–455.
5. Шеер А.В. Бизнес-процессы. Основные понятия. Теория. Методы : пер. с англ. — 2-е изд., испр. и доп. — М. : Просветитель, 1999. — 152 с.
6. Коннов И.В. Применение ослабленных отношений симуляции в методе сетевых инвариантов для верификации параметризованных асинхронных моделей // Моделирование и анализ информационных систем. — 2010. — Т. 15, № 3. — С. 3–13.
7. Hesuan Hu, MengChu Zhou, ZhiWu Li. Algebraic Synthesis of Timed Supervisor for Automated Manufacturing Systems Using Petri Nets // Automation Science and Engineering IEEE Transactions on. — 2010. — Vol. 7, № 3. — P. 549–557.