

## ПОДГОТОВКА ПРОВЕДЕНИЯ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА С ПОМОЩЬЮ МЕТАКОМПИЛЯЦИИ НА ПРИМЕРЕ SHAREPOINT

Н.Н. Войт<sup>18</sup>, С.Ю. Кириллов<sup>19</sup>, В.А. Гордеев<sup>20</sup>, Д.С.  
Канев<sup>21</sup>

В статье рассматривается проблема нехватки данных для проведения вычислительного эксперимента для оценки эффективности контроля диаграмматических моделей потоков работ с помощью разработанного анализатора. Предлагается использовать метакомпиляцию RVTI-грамматики для последующего порождения нужных цепочек, тем самым автоматизируя процесс получения тестовых данных. Приводятся примеры на основе описания визуального языка SharePoint.

### Введение

В жизненном цикле большинства серьезных проектов не обходится без такой стадии как тестирование получившегося решения. Так происходит и с разрабатываемом моей научной группой средством обработки графических моделей, где под обработкой подразумевается контроль, анализ и трансляция диаграмм. После разработки возникла необходимость проведения эксперимента для подтверждения корректности работы анализатора и его предполагаемых теоретических характеристик.

Был успешно применен метод ручного тестирования. Диаграммы разрабатывались специалистами, знающими SharePoint. Также была собрана база из открытых источников в сети интернет. Дополнительно к работе были подключены студенты. К сожалению, ручной метод требовал много человеческих усилий и времени, а в случае со студентами было малое количество корректных диаграмм, связанное с плохим знанием SharePoint. Поэтому стоит говорить о проблеме нехватки тестовых данных.

---

<sup>18</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru

<sup>19</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: kirillovsiyu@gmail.com

<sup>20</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: v.gordeev@ulstu.ru

<sup>21</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: dima.kanev@gmail.com

Для решения проблемы необходимо автоматизировать процесс генерации тестовых данных. Предполагается, что полученные данные будут соответствовать определенным критериям, которые позволят провести тестирование с замером времени выполнения обработки диаграмм, выявлением количества положительных и ложных срабатываний анализатора на ошибках различных типов.

## **Обзор**

Приведем определения позитивного и негативного тестов.

Позитивный тест – последовательность квазитермов (диаграмма), являющаяся предложением целевого диаграммного языка. Негативный тест – последовательность квазитермов (диаграмма), не являющаяся предложением целевого диаграммного языка.

Используя RVTI-грамматику можно не только проверить диаграмму на корректность, но и с помощью неё породить новую. Построение правильных диаграмм по RVTI-грамматике описана в работе [1] и не требует дополнительных решений. В настоящее время существует немалое количество методов для генерации негативных диаграмм (тестов). Порождение негативных диаграмм описано в работах [2,3]. Подход верен, но достаточно трудоемок.

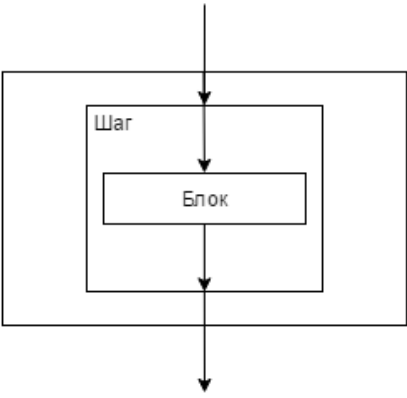
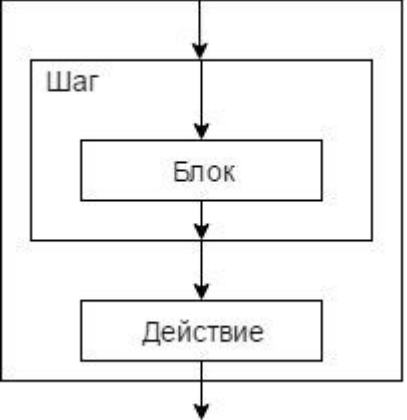
Одним из основных методов генерации негативных диаграмм, предложенный в статье [4], является так называемый метод мутационного тестирования. Основой данного метода является предположение о том, что при изменении исходной грамматики языка, ее можно использовать в качестве потенциального источника негативных диаграмм.

## **Использование метакомпиляции для генерации порождающей RV-грамматики**

В данной статье для генерации негативных диаграмм используется метакомпиляция, а именно синтезируются грамматики описывающие целевой язык заведомо неверно. Такой подход позволяет генерировать такую негативную грамматику, которая необходима пользователю. Так как пользователь сам задает ошибки в правилах описания грамматики это позволяет контролировать весь процесс создания негативной грамматики.

Идея метода в том, что в корректные правила описания вносятся изменения и на основании них метакомпилятор формирует негативную RVTI-грамматику. Один из способов — это соединить несоединяемые блоки. Пример приведен в таблице 1.

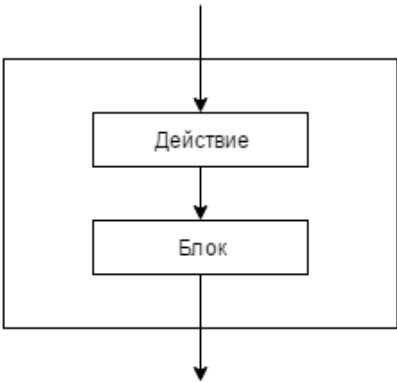
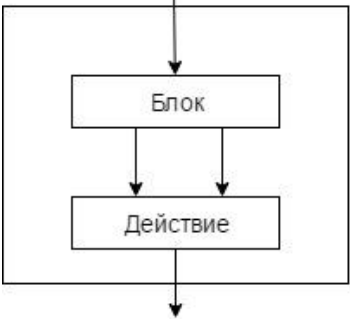
Таблица 1. Пример ошибочного правила

Графическое представление	Текстовое представление
	<p><b>Rule:</b> IdSteps  <b>Consist of:</b> IdBeginStep                  EIBeginStep, IdBlock EIBlock,                  IdEndStep EIBlock</p> <p><b>Internal Relationships:</b>                  EIBlock.In = EIBeginStep.Out,                  EIBlock.Out = EIBlock.In</p> <p><b>External Relationships:</b>                  IdSteps.In = EIBeginStep.In,                  EIBlock.Out = IdSteps.Out;</p>
	<p><b>Rule:</b> IdSteps  <b>Consist of:</b> IdBeginStep                  EIBeginStep, IdBlock EIBlock,                  IdEndStep EIBlock, IdAction                  EIAction</p> <p><b>Internal Relationships:</b>                  EIBlock.In = EIBeginStep.Out,                  EIBlock.Out = EIBlock.In,                  EIBlock.Out = EIAction.In</p> <p><b>External Relationships:</b>                  IdSteps.In = EIBeginStep.In,                  EIAction.Out = IdSteps.Out;</p>

При формировании грамматики ошибок обнаружено не будет, но элемент «Действие» не может находиться за пределами элемента «Шаг» и это будет являться ошибкой, хотя сформированная RVTI-грамматика не выведет ошибки. Такие ошибки легко формируются и таким образом, можно задать абсолютно любые сочетания элементов.

Следующий вариант — это изменение количества входов/выходов в элементах. С помощью изменения правил, можно задать абсолютно любое количество входов/выходов, что позволит модифицировать элементы. Пример изменения описания в таблице 2.

Таблица 2. Пример ошибочного правила

Графическое представление	Текстовое представление
	<p><b>Rule:</b> IdBlock  <b>Consist of:</b> IdAction                      EAction, IdBlock EBlock  <b>Internal Relationships:</b>                      EBlock.In = EAction.Out  <b>External Relationships:</b>                      EBlock.In = EAction.In,                      IdBlock.Out = EBlock.Out</p>
	<p><b>Rule:</b> IdBlock  <b>Consist of:</b> IdAction                      EAction, IdBlock EBlock  <b>Internal Relationships:</b>                      EBlock.In1 = EAction.Out1,                      EBlock.In2 = EAction.Out2  <b>External Relationships:</b>                      EBlock.In = EAction.In,                      IdBlock.Out = EBlock.Out</p>

Например, можно добавить еще одну входную связь к элементу «Действие». При таком изменении исходного правила описания конструкции языка, полностью изменится грамматика элемента, что позволит легко его сгенерировать. При этом такой случай будет ошибочным, хотя при метакомпиляции ошибок не произойдет, и RVTI-грамматика не будет находить текущую ошибку.

При таком подходе пользователю доступны все возможные изменения грамматики и при правильном использовании, можно задать абсолютно любую ошибку, которая будет генерироваться. Это позволит проводить тестирование быстрее и с большей точностью.

### Заключение

Автоматизация генерации тестовых цепочек позволит помочь в проведении наиболее корректного вычислительного эксперимента для

оценки эффективности контроля диаграмматических моделей потоков работ с помощью разработанного авторами анализатора на базе RVTI-грамматики. А метод на основе метакомпиляции устраняет большое количество ручного труда авторов по перестраиванию порождающего автомата, что без сомнений является его огромным преимуществом.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417. Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 18-47-730032. Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

### Список литературы

1. Афанасьев, А.Н., Кириллов, С.Ю. (2016) Разработка генератора правильных BPMN диаграмм на основе RV-грамматики. Сборник научных трудов VIII Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых "Информатика и вычислительная техника" (ИБТ 2016), 47-55.
2. Афанасьев, А.Н., Кириллов, С.Ю., Воеводин, Е.Ю. (2016) Разработка генератора негативных BPMN диаграмм на основе RV-грамматики. Сборник научных трудов VIII Всероссийской школы-семинара аспирантов, студентов и молодых ученых "Информатика, моделирование, автоматизация проектирования" (ИМАП-2016), 247-254.
3. Афанасьев, А.Н., Кириллов, С.Ю., Савичева, Ю.А. (2016) Разработка динамического генератора негативных BPMN диаграмм на основе RV-грамматики. Сборник научных трудов VIII Всероссийской школы-семинара аспирантов, студентов и молодых ученых "Информатика, моделирование, автоматизация проектирования" (ИМАП-2016), 255-263.
4. Maurer P.M., "The design and implementation of a grammar-based data generator," Software: Practice and Experience, Vol. 22, No. 3, 1992. pp. 223–244