

Отклики модели: *servedAuto* – количество обслуженных автомобилей; *commonQueueTime* ~ среднее время ожидания в общей очереди; *commonQueueLength* – средняя длина общей очереди; *stationQueueTime* – среднее время ожидания в очередях перед колонками; *stationQueueLength* – средняя длина очередей перед колонками; *operatorQueueTime* – среднее время ожидания в очереди к оператору; *operatorQueueLength* – средняя длина очереди к оператору; *kLoadOperator* – коэффициент загрузки оператора.

Имитационная модель реализована с помощью системы моделирования MICIC4. Результаты верификации обсуждаются в докладе.

**С.В. Леванцов** (УО «ГГТУ им. П.О. Сухого», Гомель)  
Науч. рук. **В.В. Комраков**, канд. техн. наук, доцент

### **КАСТОМИЗАЦИЯ СЕРВИСОВ ПРИЛОЖЕНИЯ TRAVELDISTRIBUTIONPLATFORMПРИ ПОМОЩИ СИСТЕМЫ BUSINESSRULESENGINE**

Задача системы – кастомизация работы сервисов, разработанных в при помощи приложения *Travel Distribution Platform*. Администратор приложения имеет доступ к различным функциям системы. На странице *BusinessModels* администратор приложения видит список всех моделей, которые присутствуют в приложении. Для каждого из сервисов представлена та же самая модель, что и описана в Java-классах и в базе данных. Однако присутствие всей модели еще не означает, что она может быть сразу же использована. Для этого нужно добавить необходимые элементы из моделей в маппинг и дать им имя. После этого у системы правил появится доступ к этим элементам по заданному имени, и появится возможность манипулирования значениями, которые будут находиться в заданных этой моделью объектах.

На странице *RuleEditor* находятся все бизнес-правила, которые есть в приложении. Стоит отметить, что бизнес-правила нужны не во всех сервисах и если мы не создали никакого маппинга для модели, то система оповестит пользователя, что для данного сервиса не может быть никаких правил, если отсутствует маппинг. В случае присутствия маппинга в модели появляется возможность написать правило, используя необходимые нам элементы модели. Все модели разбиты на категории, каждая категория содержит как минимум 1 правило. Каждое правило имеет собственное имя. Разбитие на категории позволяет вызвать только определенный список правил в конкретный момент времени, а не вызывать все сразу.

Сами правила представляют собой псевдокод, который в последствии превращается в xml объект, из которого сгенерируется Java-класс, который будет добавлен в уже развернутое на сервере приложение.

Правила имеют следующие конструкции:

- 1 Оператор условия If-then.
- 2 Оператор цикла Foreach.
- 3 Оператор удаления объектов Remove.
- 4 Оператор добавления объектов Add;
- 5 Логические операторы true/false.
- 6 Операторы проверки на нулевую ссылку null/isnonnull.
- 7 Оператор проверки вхождения значения элемента в перечисление доступных элементов In.
- 8 Алгебраические операторы +, -, /, \*.
- 9 Оператор присваивания значения переменной Assign.
- 10 Функция копирования элемента или блока операций Copy.
- 11 Функция удаления элемента или блока операций Delete.

Тип данных в правилах нестрогий. Мы можем объявить примитив числового типа, а ниже в правиле его переопределить в логический тип данных. Также в правиле не существует понятия области видимости переменной, это на практике означает, что мы в цикле объявили некую переменную, то система правил без каких-либо проблем разрешает ее дальнейшее использование вне цикла.

Что касается типов данных, то они разделены на отдельные группы:

- 1 Literal – содержит в себе логический тип данных, дату, время, а также обычную строку.
- 2 Mappedelement – содержит в себе тот тип данных, которым обладает элемент из выбранной нами модели.
- 3 Parameter – переменные данного типа берут свои значения из таблиц, описанных в DataEditor;
- 4 Variable – это меняющийся тип данных, в зависимости от присваиваемого значения, по умолчанию это строка.

На странице *DataEditor* находятся таблицы с пользовательскими данными, относящимися к конкретным правилам. Данные таблиц экспортируются в формате *csv*, которые в последствие будут использованы правилом. За каждым правилом закреплена своя таблица, и правила не могут использовать данные других таблиц.

На данной странице присутствуют следующие возможности:

- 1 Не редактируемый просмотр правила.
- 2 Добавление элементов в таблицу.
- 3 Оператор удаления объектов Remove.
- 4 Оператор добавления объектов Add.