поддерживается. К серверным языкам программирования можно отнести: PHP, Perl, Python, Ruby, любой .NET язык программирования (технология ASP.NET), Java, Groovy.

Важной стороной работы серверных языков является возможность организации непосредственного взаимодействия с системой управления базами данных (или СУБД) – сервером базы данных, в которой упорядоченно хранится информация, которая может быть вызвана в любой момент.

Е.В. Леванцов (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.Н. Леванцов**, старший преподаватель

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ АВТОЗАПРАВОЧНОЙ СТАНЦИИ

Объект моделирования представляет собой автозаправочную станцию, предназначенную для заправки автомобилей топливом. На станции установлено равное количество колонок для автомобилей с левосторонним и правосторонним баком. Каждая колонка может обслуживать только один автомобиль. Но перед ней имеется место для ожидания некоторого числа автомобилей. Автомобили, которым не хватило места ни в одной очереди перед подходящей колонкой, ожидают в общей очереди до тех пор, пока хотя бы одно такое место не освободится.

Процесс обслуживания на станции осуществляется следующим образом. Когда очередной автомобиль подъезжает для заправки, он становится в общую очередь. Как только появляется (или уже существует) свободное место в очереди перед какой-либо подходящей колонкой, автомобиль занимает это место и ждет обслуживания там. Как только эта колонка освобождается, автомобиль подъезжает к колонке, но перед началом заправки топливом водитель идет оплачивать его у оператора. Если оператор занят, водитель становится в очередь. Далее происходит расчет с оператором, сразу после которого начинается заправка. По ее окончанию автомобиль уезжает.

Параметры модели: Tap – среднее время между появлениями автомобилей на автозаправке; Dap – дисперсия этого времени; Pleft – вероятность того, что появившийся автомобиль имеет бак слева; N – количество колонок для автомобилей с определенным видом бака; M – количество мест для ожидания автомобилей перед каждой из колонок; Top – среднее время расчета с оператором; Bop – дисперсия этого времени; Tf – среднее время заправки топлива в один автомобиль; £>/-дисперсия этого времени.

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.