

другом месте кода. Сеть же может быть реализована и для значительно большего количества мостов (т.е. сотни потоков). Автором предлагается упрощенная модель сети Петри, предусматривающая наличие в позиции одной фишки (объект TPosition). Топология Петри реализуется при помощи массива базового класса TBaseState и матрицы, которая определяет с чьими выходами соединены входы элемент (мост) TBridge. Все классы являются наследниками потокового класса TThread и выполняют в потоке проверку изменения состояния входов и установку выходов. Выделенный мост - класс – TFinalBridge выполняется в потоке с низким приоритетом и при активации моста осуществляет переход на защищаемую процедуру, что возможно при определенном начальном состоянии сети Петри.

Литература

1. Защита программ с помощью сетей Петри. – Режим доступа: <http://www.wasm.ru/print.php?article=petri>. – Дата доступа: 18.05.2004.
2. Котов, В.Е. Сети Петри / В.Е. Котов. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.

СОЗДАНИЕ АЈАХ-ПРИЛОЖЕНИЯ

Савлук Е. В.

ГрГУ им. Янки Купалы, Гродно, Беларусь, e-mail: kate90s@mail.ru

Технология web-программирования АЈАХ – это подход к построению интерактивных пользовательских интерфейсов web-приложений, который заключается в фоновом обмене данными браузера с сервером. Это значит, что приложение выполняет большую часть работы на стороне клиента и взаимодействует с сервером посредством внеполосных обращений. В результате web-страницы динамически обновляются без полной перезагрузки, прерывающей обмен данными. Поэтому с помощью АЈАХ можно создавать более быстрые, гибкие и удобные пользовательские интерфейсы для web-приложений. АЈАХ-приложения пишут, как правило, на JavaScript и для передачи данных используют объект XMLHttpRequest, который был одобрен W3C. Технология получила широкое распространение относительно недавно, поэтому следует учитывать проблему кроссбраузерности.

Для демонстрации возможностей и исследования аспектов применения АЈАХ в web-программировании, ставится задача создать web-приложение «Расписание занятий». На АЈАХ выполняется вывод дополнительных полей формы при выборе пользователем одного из пунктов «для студента», «для преподавателя» или «для аудитории». Без использования технологии АЈАХ, браузеру пришлось бы выполнить полную перезагрузку страницы при этом действии пользователя, вместо того, чтобы просто добавить несколько дополнительных элементов формы. Также на АЈАХ реализуется заполнение поля «Группа», которое должно быть заполнено лишь группами, которые учатся на выбранном факультете на определенном курсе. Следовательно, любое изменение в списках «Факультет» и «Курс» будет вызывать функцию, которая заполнит список групп. Очевидно, что без АЈАХ данный сценарий был бы, попросту, невозможен, и пришлось бы искать другой, не столь красивый и удобный, выход из ситуации.

Но, как и у всякой технологии, у AJAX есть свои недостатки. Во-первых, динамически загружаемое содержимое недоступно поисковикам. Во-вторых, динамически создаваемые страницы не регистрируются браузером в истории посещения страниц, поэтому не работают стандартные инструменты браузера. И наконец, усложняется логика проекта, и эффект технологии может быть нивелирован необоснованным ростом затрат на создание приложения. Однако следует понимать, что эти недостатки возникают, прежде всего, из-за нерационального использования технологии. Применение AJAX оправдано только тогда, когда это действительно позволит оптимизировать работу приложения.

Литература

1. Бибо, Б. Аjax на практике (Ajax in Practice)/ Б. Бибо, Д. Крейн, Дж. Сонневельд. – М.: Вильямс, 2007. – 680с.
2. Маклафлин, Б. Изучаем Ajax (Head Rush Ajax)/ Б. Маклафлин. – СПб.: Питер, 2007. – 460с.

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА «ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ». МОДУЛЬ «СОТРУДНИК»

Синяков В. М.

ВГУ им. П. М. Машерова, Витебск, Беларусь, e-mail: vsu@vsu.by

В рамках реализации проекта «Электронный университет» был разработан и реализован модуль «Сотрудник», включающий в себя функционал по управлению кадрами университета, а именно:

1. Разработана личная карточка сотрудника университета.
2. Разработана система управления контрактами сотрудников (см. рис. 1). Реализована история контрактов сотрудника в его личной карточке.
3. Разработаны система управления отпусками сотрудников и календарь отпусков. Реализована история отпусков сотрудника в личной карточке.
4. Разработан функционал, позволяющий управлять надбавками, наградами, историей повышения квалификации сотрудника.
5. Разработана гибкая система управления штатным расписанием университета, с возможностью печати штатного расписания.

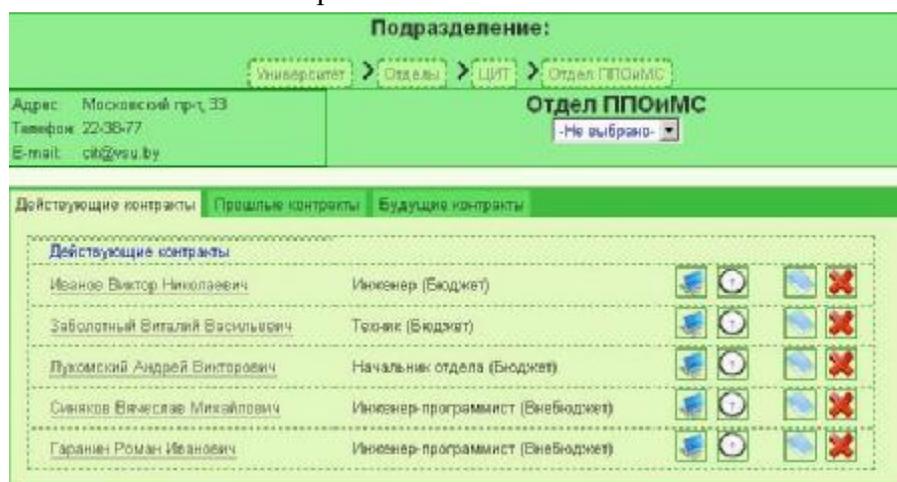


Рис. 1. Система управления контрактами