Чтобы надежно застраховать web-приложение от SQL-инъекций необходимо выполнение следующих принципов:

1 – работа с СУБД в приложении должна осуществляется не напрямую, а через специальную *«прослойку»*, предоставляющую соответствующий интерфейс;

2 – *любое* динамическое включение данных в SQL-запрос, должно происходить под контролем данной *«прослойки»*.

Подробности реализации данных принципов на практике рассматриваются в докладе.

Так как в больших проектах всегда есть вероятность случайного отклонения от данного метода работы с СУБД, автор предлагает создать специальную программу, которая проверяет код на соответствие принципам безопасной работы с СУБД. Принцип работы и детали реализации данной программы освещаются в докладе.

Преимущество данного подхода в том, что он позволяет на сто процентов устранить все SQL-инъекции уже на стадии реализации приложения, а не на стадии тестирования.

ПРОБЛЕМЫ СОВРЕМЕННЫХ CMS/CMF И ПУТИ ИХ РЕШЕНИЯ

А.И. Хобня, О.М. Демиденко
(*ГГУ им. Ф. Скорины, Гомель*)

Рассмотрев и проанализировав наиболее мощные и популярные системы, позиционируемые их разработчиками как CMS/CMF, т.е. сочетающие в себе функции фрэймворка для разработки web-приложений (CMF – Content management framework) и готовой системы управления содержимым (CMS – Content management system), можно выделить несколько самых существенных и присущих многим из данных систем недостатков:

– сложность для конечного пользователя и низкий уровень системы администрирования;

– проблемы наличия шаблонов;

– непрозрачная, местами запутанная, местами избыточная архитектура, и как следствие либо сложности при создании модулей и необходимость править код ядра системы, либо сложность в освоении и правильном понимании архитектуры системы для разработчика, что

ведет к написанию «неправильных» модулей, плохо влияющих на работу остальных частей системы;

– низкая производительность;

– отсутствие интегрируемости поставляемых WYSIWYG-редакторов с дизайном сайта и семантикой содержимого;

– недостаточная, запутанная или громоздкая и избыточная документация.

Разрабатывая собственную CMS/CMF, предлагается избежать перечисленных выше недостатков следующими путями:

– создание фрэймворка для построения панелей администрирования;

– создание гибкого, простого, но быстрого шаблонизатора;

– создание прозрачной логичной архитектуры с использованием ООП;

– реализация двух направлений в повышении производительности: оптимизация работы с файлами и базой данных, создание гибкой многоуровневой системы кэширования;

– создание WYSIWYG-редактора, интегрируемого с дизайном сайта через использование CSS и генерирующего чистый семантический HTML(XHTML)-код;

– разработка единого стандарта документации, наиболее подходящего для проекта такого типа.

Детали реализации CMS/CMF рассматриваются в докладе.

РАБОТА С СИСТЕМОЙ ВИЗУАЛИЗАЦИИ ТРЁХМЕРНОЙ ГРАФИКИ OGRE3D

И.Н. Шавловский, В.С. Давыдов
(ГГУ им. Ф. Скорины, Гомель)

Концепция объектно-ориентированного программирования появилась ещё в 1967 году – именно тогда был создан первый объектно-ориентированный язык Симула. Основным преимуществом ООП является удобство в работе, а также систематический подход к разработке приложений.

В 2005 году, программист Стив Стритинг (Steve Streeting) представил миру объектно-ориентированый движок для визуализации трёхмерной графики OGRE3D (Object-Oriented Graphics Rendering Engine), который, оказался очень мощным и гибким средством для визуализа-