

УДК 004

“SMART” ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА С ИНКАПСУЛИРОВАННЫМ SQL-ЗАПРОСОМ

Марулин С.Ю., Ахмет Алай

к.т.н., ст. преподаватель кафедры системного программного обеспечения Марулин С.Ю.
Одесский Национальный Политехнический Университет, УКРАИНА

АННОТАЦИЯ. В тезисах описан поход автоматизированного создания интерфейса доступа к источнику данных, реализующий парадигму повторного использования кода и метапрограммирования.

Введение. Одной из самых сложных задач *project manager'a* – с минимальным затратами временных, финансовых и человеческих ресурсов реализовывать программные проекты. Компании по разработке программного обеспечения (ПО), разрабатывающие большое количество ПО как правило используют подход повторного использования кода и тем самым снижают трудозатраты и время на разработку нового продукта. Более того, если код повторно использован, то возможно, что он хорошо протестирован и, вероятно, не содержит “багов”.

Таким образом, повторное использование кода (*code reuse*) – **методология проектирования** ПО, заключающаяся в том, что система частично либо полностью должна состояться из частей, написанных ранее компонентов и/или частей другой системы, и эти компоненты должны применяться более одного раза [1].

Проблема повторного использования кода. Использование написанного фрагмента при разработке и проектировании нового ПО вносит преимущества в новый проект, но в тоже время и несет скрытые неожиданности: ошибки и несоответствия текущим потребностям заказчика.

Повторное использование кода относится к парадигме **метапрограммирования** [2] и требует от команды программистов определенной культуры и квалификации. Высококвалифицированный программист должен обладать не только навыками обычного кодирования, но разбираться в высокоуровневых спецификациях. Создавать “правильный” код для повторного использования – “убивать одним выстрелом двух зайцев” – зарабатывать на разработанном ПО и на продажи его модулей. Подходы, автоматизирующие процесс создания ПО и использующие при этом технику метаописаний лежат также и в основе технологий автоматизированного создания элементов управления (ЭУ) пользовательского интерфейса [3].

В литературе описываются множество подходов к адаптиванию различных ЭУ пользовательского интерфейса к экранам различных девайсов, большое внимание уделяется эргономике и дизайну, но не описываются подходы к созданию ЭУ со встроенной логикой (например, в виде *SQL*-запросов).

Учитывая все достоинства и недостатки парадигмы метапрограммирования, предложена технология автоматизированного создания экранных форм пользовательского интерфейса (ЭФПИ) на базе методики повторного использования кода с элементами декларативного описания *SQL*-запросов, как источников данных для ЭУ.

На рисунке 1 представлена функциональная модель данной технологии “*SmartU*”.

Предложенная технология использует в качестве языка программирования - язык *Java*, в качестве источника данных СУБД *PostgreSQL* и состоит из следующей последовательности:

- “Создать *SmartU* ЭУ”;
- “Установить имя *SmartU* ЭУ”;
- “Инкапсулировать *SQL*-запрос в *SmartU* ЭУ”;
- “Связать *SmartU* ЭУ”;
- “Выстроить зависимости “*SmartU*” ЭУ”.

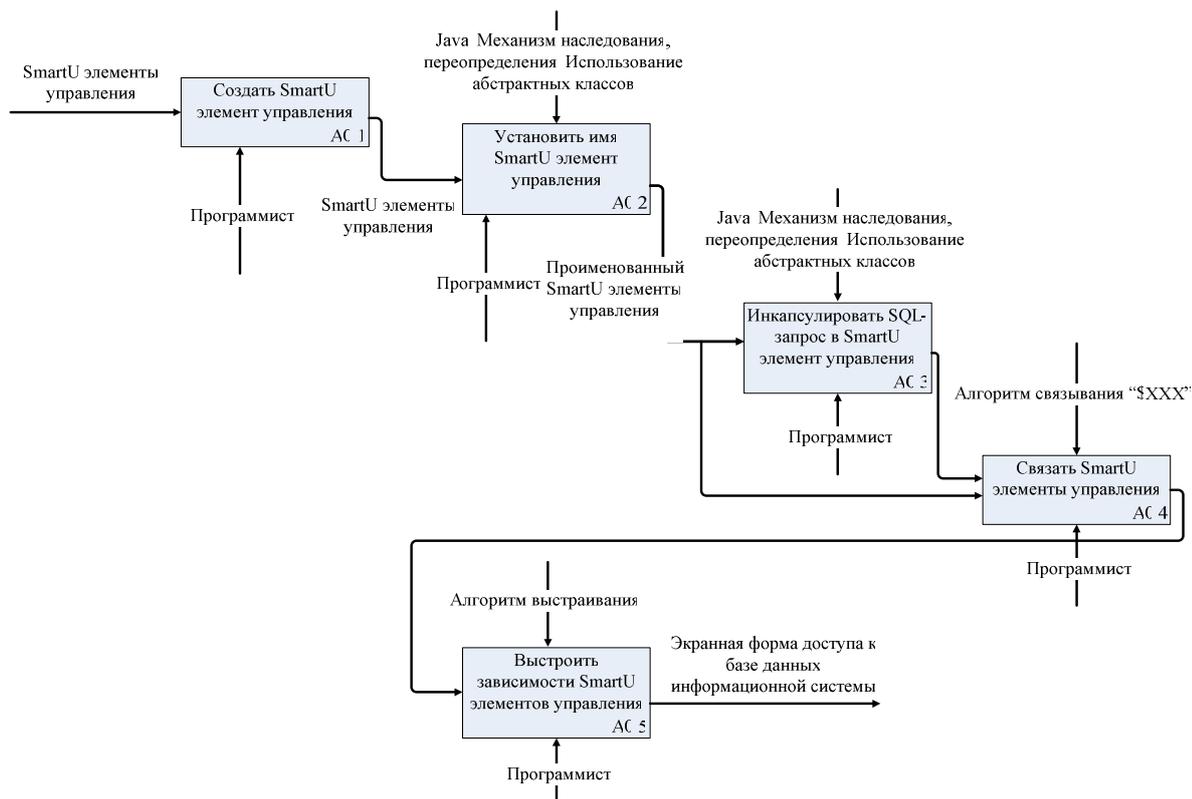


Рисунок 1 – Функциональная модель технологии “SmartU”

Основой предложенного подхода является алгоритм связывания ЭУ между собой. Такой алгоритм позволяет с помощью параметризованного *SQL*-запроса связывать между собой ЭУ. Параметризованный *SQL*-запрос содержит шаблонные значения типа *name[digit]*, где *name* – имя элемента управления, *digit* – порядковый номер поля таблицы записей, которую возвращает запрос, например:

```
select detailed_plan_id, cathedra_own_id, teach_subject
from view_detailed_plan
where teach_year_id = $teach_year[0] and
group_id = $group[0]
```

Для создания интерфейса доступа к источнику данных программисту необходимо с помощью стандартной конструкции *Java* – *Class obj = new Class()* создать массив ЭУ и параметризовать *SQL*-запросы этих элементов.

Расчет показателя эффективности работы технологии “SmartU”. Как уже упоминалось во введение основное преимущество повторного использования кода – уменьшение времен на разработку ПО. Исходя из этого, был проведен эксперимент и рассчитаны два показателя времени: без использования предложенного подхода – ручное программирование и с применением библиотеки “SmartU”.

Показатели времени необходимые для создания ЭФПИ при ручном режиме и с использованием библиотеки “SmartU” были подсчитаны относительно показателя экспертного времени. Экспертный показатель времени – показатель определенный группой программистов, разрабатывающих подобные экранные формы. Данный показатель отображает среднее время необходимое на разработку “среднесложной” ЭФПИ.

Полученные значения позволили сделать вывод об эффективности использования технологии “SmartU” и выигрыш во времени 71%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. James W. Hooper and Rowena Chester. Software Reuse - Guidelines and Methods. Plenum Press, New York, 1991.
2. Джонатан Бартлетт, Искусство метапрограммирования, Часть 1: Введение в метапрограммирование.
3. Basnyat S., Bastide R., Palanque P.: Extending the Boundaries of Model-Based Development to Account for Errors. MDDAUI '05. 2005.