

Томский государственный университет
Кемеровский государственный университет
Кемеровский научный центр СО РАН
Институт вычислительных технологий СО РАН
Филиал Кемеровского государственного университета
в г. Анжеро-Судженске

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И МАТЕМАТИЧЕСКОЕ
МОДЕЛИРОВАНИЕ
(ИТММ-2011)**

**Материалы X Всероссийской
научно-практической конференции
с международным участием
25–26 ноября 2011 г.
Часть 1**

Издательство Томского университета
2011

УДК 519

ББК 22.17

И74

Редакционная коллегия:

Р.Т. Якупов, д-р физ.-мат. наук, профессор;

А.А. Назаров, д-р техн. наук, профессор;

И.Р. Гарайшина, канд. физ.-мат. наук, доцент

И74 Информационные технологии и математическое моделирование
(ИТММ-2011): Матер. X Всерос. науч.-практ. конф. с междунар.
участием (25–26 ноября 2011 г.). – Томск: Изд-во Том. ун-та, 2011.
Ч. 1. 192 с.

ISBN 978-5-7511-2031-3

В часть 1 вошли материалы секций «Информационные технологии» и
«Вероятностные методы и модели».

Для специалистов в области информационных технологий и математи-
ческого моделирования.

УДК 519

ББК 22.17

ISBN 978-5-7511-2031-3 © Томский государственный университет, 2011

© Кемеровский государственный университет, 2011

© Кемеровский научный центр СО РАН, 2011

© Институт вычислительных технологий СО РАН, 2011

© Фил-л Кемеровского государственного университета
в г. Анжеро-Судженске, 2011

*Конференция проводится при поддержке Российского фонда фунда-
ментальных исследований (проект № 11-07-06076-г)*

Кроме того существуют трудности в управлении проектами, поскольку участники проекта (заказчики, разработчики, исполнители) находятся в разных странах.

Теневая экономика. Возвращаясь к проблемам законодательства в области ИТ-бизнеса можно говорить о том, что фирмы, занимающиеся оффшорным программированием, выводят свою деятельность за рамки закона, то есть действуют на «сером» рынке. Зачастую исполнителями заказов являются физические лица, «работающие на дому», официально не зарегистрированные, и, соответственно, не выплачивающие налоги. То же самое можно сказать и о мелких фирмах.

Внешние психологические. Расширению экспорта мешает недоверие зарубежных партнеров к российской отрасли ИТ. Мнение о нестабильности таможенного налогового законодательства, недостаточном знании английского языка, отсутствии контроля качества в российских компаниях увеличивает количество претензий к защите прав на интеллектуальную собственность.

Фирмам, осуществляющим деятельность в области оффшорного программирования, сложно повлиять самостоятельно на перечисленные выше проблемы. Поэтому для максимального использования факторов роста необходимо осуществление государством мероприятий, направленных на поддержку и развитие информационных технологий в России, и совершенствование законодательной базы.

ИСПОЛЬЗОВАНИЕ МЕХАНИЗМА ДОПОЛНИТЕЛЬНЫХ МОДУЛЕЙ ПРИ ПОСТРОЕНИИ РАСШИРЯЕМОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

К.Ю. Войтиков¹, А.Н. Моисеев², П.Н. Тумаев¹

*¹Филиал Кемеровского государственного университета
в г. Анжеро-Судженске*

²Томский государственный университет

При разработке современных информационных систем, в частности системы имитационного моделирования процессов массового обслуживания [1], зачастую возникает задача обеспечения возможности ее расширения новыми алгоритмическими конструкциями в процессе эксплуатации. Стоит отметить, что эти конструкции должны быть доступны для выбора и задания параметров в визуальных средствах приложения.

Например, если рассматривать задачу моделирования случайных величин, для которой применяются специальные алгоритмы имитационного моделирования, существует множество различных законов распределения таких величин, и, более того, исследователь может конструировать собственные специфические законы (алгоритмы моделирования). Поэтому для подобного приложения не представляется возможным заранее предусмотреть и реализовать всю необходимую алгоритмическую базу, в связи с чем возникает задача обеспечения системы возможностью динамического

расширения приложения алгоритмами предметной области. Для решения этой задачи предлагается предусмотреть в системе возможность подключения пользовательских динамических библиотек, реализующих определенные интерфейсы.

Данный подход дает следующие преимущества:

- отсутствие жестких ограничений на используемый пользователем язык программирования;
- позволяет создавать сопутствующие объекты, реализация средств конструирования которых в программе является слишком трудоемкой задачей (речь идет о визуальных элементах, позволяющих задавать параметры соответствующих невизуальных конструкций, например, для законов распределения, эти параметры могут отличаться количеством, типом, способом задания и т. д.).

Предоставив пользователю возможность самостоятельно конструировать элементы управления для настройки параметров, добавляемых им к программе, отпадет необходимость не только в интерпретации программного кода внутри программы, но и потребность в визуальных средствах конструирования.

Для обеспечения расширяемости воспользуемся архитектурным решением «Дополнительный Модуль» [2]. Согласно данному подходу выделим базовый интерфейс создаваемых объектов `Object` (см. рис. 1). В общем случае мы его не конкретизируем, т.к. эти объекты имеют интерфейс, специфический для конкретной предметной области. Основное отличие предлагаемого подхода от классического [2] является то, что предлагается реализовать фабрику `Factory` как объект-коллекцию для загружаемых из дополнительных модулей типов. Чтобы обеспечить наибольшую независимость от особенностей языка программирования вводятся специальные объекты-создатели `Creator` [3], единственные экземпляры которых (паттерн Одиночка [4]) и будут содержаться в указанной коллекции. Главное и единственное назначение конкретных создателей `ConcreteCreator` – инстанцирование соответствующего класса `ConcreteObject` из текущей динамической библиотеки.

Отличительной особенностью рассматриваемой задачи является наличие второй динамически загружаемой составляющей – визуальных объектов, предназначенных для отображения и редактирования в пользовательском интерфейсе атрибутов соответствующих объектов `Object`. Устройство этой части выглядит аналогичным образом: фабрика `VisualFactory` отвечает за загрузку из библиотек и содержит коллекцию создателей с интерфейсом `VisualCreator`. Объекты `ConcreteVisualCreator`, реализующие этот интерфейс, в свою очередь, отвечают за инстанцирование соответствующих визуальных объектов `ConcreteControl`, реализующих общий интерфейс `Control`.

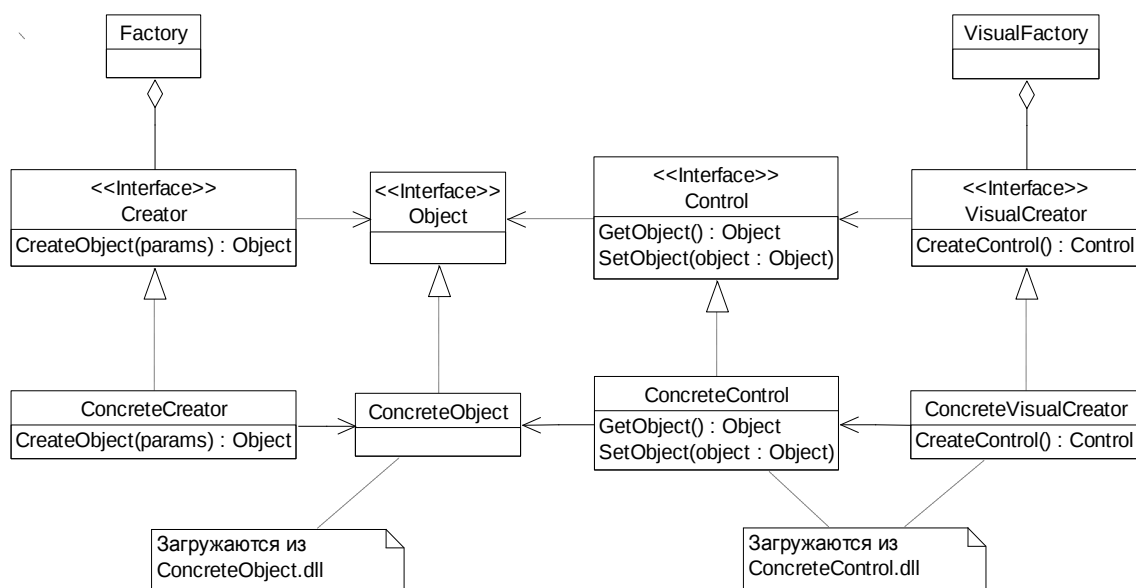


Рис. 1. Общая архитектура подсистемы подключаемых модулей

Стоит отметить, что визуальная часть архитектуры имеет следующие отличительные особенности:

- интерфейс конкретизирован до операций, выполняющих генерацию невидимого объекта на основе данных, введенных пользователем, и установку состояния визуальных элементов в соответствии со значениями атрибутов конкретного невидимого объекта (операции `GetObject()` и `SetObject()`, соответственно).
- программа может поддерживать несколько типов интерфейса пользователя, каждый из которых предполагает использование специфических групп визуальных элементов. Для реализации подобной гибкости можно воспользоваться паттерном проектирования «Абстрактная Фабрика» [4].

Таким образом, в работе предложен эскиз архитектуры приложения с расширяемой алгоритмической базой предметной области, на основе которой было реализовано программное приложение для задачи, описанной в [1].

Работа выполнена в рамках АВЦП «Развитие научного потенциала высшей школы (2009–2011 гг.)», проект № 11803.

Литература

1. Войтиков К.Ю., Моисеев А.Н., Тумаев П.Н. Компонентная модель распределенной объектно-ориентированной системы имитационного моделирования // Вестн. Том. гос. ун-та. Управл., вычисл. техн. и информ. 2010. № 1. С. 78–83.
2. Фаулер М. Архитектура корпоративных программных приложений / Пер. с англ. М.: Вильямс, 2006. 544 с.
3. Ларман К. Применение UML и шаблонов проектирования. 2-е изд. М.: Вильямс, 2004. 624 с.
4. Гамма Э., Хелм Р., Джонсон Р., Влессидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2010. 368 с.