МЕТОД ИНТЕГРАЦИИ ВНЕШНИХ КОМПОНЕНТ В ИНСТРУМЕНТАЛЬНУЮ СИСТЕМУ В УСЛОВИЯХ ОГРАНИЧЕННЫХ ВОЗМОЖНОСТЕЙ МОДИФИКАЦИИ ИСХОДНОГО КОДА

СКОРЛЫШЕВ Д.Ю., ШКИРТИЛЬ В.И.

УДК 004.4'23

Скорлышев Д. Ю., Шкиртиль В. И. **Метод интеграции внешних компонент в инструментальную систему в условиях ограниченных возможностей модификации исходного кода.**

Аннотация. В статье предлагается метод интеграции внешних компонент в инструментальную систему в условиях ограниченных возможностей модификации исходного кода. Описываются основные характеристики интегрируемых компонент, а также особенности процесса их интеграции, требующие поиска новых методов объединения инструментальных систем. В статье анализируются различные методы интеграции информационных систем. Описывается предлагаемый метод интеграции за счет обеспечения доступа к внутренним обработчикам интегрирующей системы и использования принципов динамического интерфейса.

Ключевые слова: интеграция, инструментальная система, обработчик, компонента.

Skorlyshev D. Y., Shkirtil V. I. **Method of external component integration to instrumental system in conditions of limited abilities of source code modification.**

Abstract. This article describes method of external component integration to instrumental system in conditions of limited abilities of source code modification. Basic characteristics of integrated components are described along with features of integration process that need new methods of instrumental system unification. Different methods of information system integration are analysed. Method of integration based on provision of access to internal handlers of host system and usage of dynamical interface principles is described.

Keywords: integration, instrumental system, callback, component.

1. Введение. Современная инструментальная система, для того чтобы бы быть востребованной, обязана предоставлять возможность дальнейшего расширения, как собственного набора функций, так и использования функций внешних компонент. В данной статье мы рассмотрим проблему интеграции внешних компонент в базовую инструментальную систему в условиях ограниченных возможностей по модификации кода как самой системы, так и интегрируемой компоненты.

Под интеграцией будем понимать добавление в основную систему (интегрирующую) внешней системы (интегрируемой). Такое объединение должно обеспечивать расширение и модификацию пользовательского интерфейса (ПИ) интегрирующей системы для обеспечения доступа к функциям интегрируемой системы и к функциям интегри-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

руемой системы путем выполнения набора консольных команд с параметрами. При написании команд могут использоваться зарезервированные слова, которые заменяются соответствующими значениями при исполнении команд. Зарезервированные слова могут использоваться как для передачи данных из системы, так и для получения данных извне.

Под интегрирующей системой будем понимать некоторую инструментальную систему, предназначенную для осуществления одного или нескольких этапов разработки и отладки программного обеспечения (ПО).

В качестве внешних интегрируемых систем — компонент, подлежащих интегрированию, будем рассматривать вспомогательные инструменты, используемые в процессе разработки ПО: компиляторы, утилиты сборки, утилиты конфигурирования, утилиты форматирования, отладчики, загрузчики, генераторы и т. д.

К интегрируемым компонентам предъявляются следующие требования:

— наличие консольного режима (режим без графического интерфейса),

— наличие параметрового запуска компоненты,

— возможность неинтерактивного режима функционирования,

— возможность приведения результатов работы к определенному виду и перенаправления их вывода.

2. Основные подходы к интеграции. Современные технологии интеграции информационных систем разделяют решаемую нами задачу на две части: 1) интеграцию данных и 2) интеграцию систем.

Для интеграции данных часто используется стандартизированный язык описания данных, как правило, использующий формат XML. Для интеграции систем обычно предлагается использовать единые технологии. У каждой компании, занимающейся разработкой ПО, они, как правило, свои: «Microsoft.net» [1], Java и т. д.

В интересующей нас области инструментальных систем это, как правило, означает интеграцию компонент, использующих некоторую конкретную технологию, что значительно ограничивает сферу их применения.

В процессе интеграции внешних компонент, построенных на различных технологиях и отвечающих заданным ограничениям, наиболее часто, используются два пограничных метода:

1) интеграция на уровне пользовательского интерфейса,

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

2) интеграция на основе использования в интегрируемой системе набора внешних функций интегрирующей системы.

3. Интеграция на уровне интерфейса. Как правило, такой подход подразумевает модификацию основного окна интегрирующей системы, добавление новых пунктов меню и кнопок на панели инструментов. Модификация обеспечивается за счет изменения базового описания интерфейса, где каждому добавляемому элементу сопоставляется консольная команда, реализующая взаимодействие с внешней компонентой.

При написании консольных команд могут быть использованы зарезервированные слова, с помощью которых обеспечивается доступ к внутренним объектам интегрирующей системы для передачи их значений в интегрируемые компоненты.

Например, зарезервированные слова могут использоваться для передачи следующих значений:

— путь к редактируемому объекту;

— путь к исполняемому файлу;

— текущее значение поля, имеющего фокус и т. д.

Редактирование интерфейса может осуществляться и на уровне описаний, и с помощью специальных графических инструментов-конструкторов.

Основными достоинствами данного способа являются простота и отсутствие необходимости в модификации исходного кода основного приложения, а также интегрируемой компоненты. К существенному же недостатку относится ограниченность функций, реализуемых данным способом интеграции, как правило, они ограничиваются функциями старт/стопа внешней компоненты с определенными параметрами и передачей в компоненту текущих значений объектов основной системы. Данные ограничения сильно снижают возможности применения описанного способа интеграции.

4. Интеграция на уровне функций. Существенно увеличить возможности по интеграции компоненты позволяет использование в ней набора внешних функций (API) основной инструментальной системы, публикуемого при запуске ее в эксплуатацию. Мощность данного API может сильно варьироваться от одной реализации инструментальной системы к другой. Теоретически API позволяет реализовывать практически полный доступ интегрируемой компоненты к механизмам основной системы.

Достоинства использования данного способа очевидны — практически любая внешняя компонента может быть органично включена

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

в основную инструментальную систему. Недостаток же один и очень существенный — интеграцию в конкретную инструментальную систему надо закладывать либо на стадии создания внешней компоненты, либо создавать некоторую программу-прослойку, оперирующую внешними API основной системы и интегрируемой компоненты.

Для решаемой нами задачи ни один из рассмотренных способов не подходит. Первый ограничивает возможности интеграции, второй требует значительной модификации интегрируемых компонент, что противоречит объявленным ограничениям.

5. Метод интеграции через доступ к внутренним обработчикам. Предлагаемое решение является модификацией способа интеграции через интерфейс, позволяющее осуществлять доступ не только к внутренним компонентам основной инструментальной системы, но и к ее внутренним механизмам, аналогично варианту использования внешнего API. Для этого предлагается использовать механизм динамического конфигурирования базового интерфейса. Под конфигурированием в данной задаче будем понимать применение некоторого конфигурационного файла-конфигурации, приводящее к изменению базового ПИ. Основной возможностью конфигурационного файла в нашем случае является возможность динамического сопоставления элементов интерфейса и внутренних обработчиков интегрирующей системы с консольными командами. Мощность языка описания конфигураций зависит от набора доступных зарезервированных команд основной системы, с помощью которых может осуществляться доступ к ее внутренним объектам с последующей передачей их значений в интегрируемые компоненты. Под доступом к внутренним обработчикам подразумевается связывание выполнения произвольной пользовательской команды (с возможностью использования зарезервированных слов) с конкретным внутренним обработчиком. Поддерживаются три варианта выполнения пользовательской команды: до, после или вместо выполнения основного обработчика. Для каждой команды указывается тип ее выполнения: асинхронно или синхронно с основной системой в зависимости от выполняемой операции. Также предоставляется возможность задания условий выполнения пользовательской команды в зависимости от значения зарезервированных слов, что расширяет возможности интеграции. При описании условия выполнения предлагается использовать скриптовый язык. Таким языком может быть, например, язык Lua, обладающий простым синтаксисом, подобным языку ANSI C [2].

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Реализация предложенного способа интеграции возможна при построении инструментальной системы с использованием динамического ПИ, когда описание всех форм и обработчиков располагается вне компилируемого исходного кода системы. Сама же инструментальная система должна содержать в себе «графическую машину», реализующую интерфейс на основе описаний [3].

Под описанием интерфейса понимается набор файлов в определенном формате, например XML, содержащих описания форм. В описании каждой формы хранится состав и расположение всех графических элементов, а также указатели на соответствующие обработчики, срабатывающие при наступлении определенных событий. Тело обработчиков располагается в компилируемом исходном коде инструментальной системы, там же производится и их регистрация. При такой реализации становится возможным расширение возможностей механизма зарезервированных слов за счет обеспечения доступа к значениям практически любых графических компонент интерфейса основной системы [4]. Таким образом обеспечивается модификация описания форм динамически — без изменения исходных кодов, за счет использования добавленных при интеграции элементов.

Инструментальная система, построенная на описанных принципах, позволяет включать вызовы интегрируемой компоненты на любом этапе функционирования основной системы, передавать данные, в том числе и введенные пользователем основной системы, и возвращать результаты работы внешней компоненты в систему.

Модификация внешней компоненты в данном варианте сводится к приведению результатов работы к стандартизированному виду, понятному интегрирующей системе, например к формату XML, и перенаправлению вывода (определенным способом именованные файлы, каналы и т. д.). Перенаправление используется для возвращения результатов работы интегрируемой системы в основную.

6. Заключение. Предложенный способ интеграции апробирован в экспериментальной инструментальной системе, предназначенной для выполнения этапов генерации шаблона исходного кода, наполнения шаблона и сборки исполняемого кода. Данная система реализована на принципах динамического интерфейса с использованием собственного языка описания форм, использующего XML-формат. В системе также реализован графический модуль конфигурирования пользовательского интерфейса, позволяющий связывать пользовательские команды с кнопками, пунктами меню и внутренними обработчиками системы.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

В качестве языка описания условий выполнения команды использован язык Lua.

Для демонстрации реализованного принципа интеграции в систему проинтегрированы:

— утилита генерации исходного кода на основании описаний элементов модели;

— утилита сравнения файлов, используемая для корректировки ошибок;

— нестандартный компилятор исходного кода.

Все интегрируемые компоненты были сторонними и не изменялись в процессе интеграции.

Интеграция включала в себя добавление пунктов меню, отвечающих за вызов приложений, добавление соответствующих кнопок на панели инструментов и внедрение вызовов компонент во внутренние обработчики системы. Так, например, на нажатие кнопки “Генератор” на панели инструментов производился вызов генератора исходного кода, который производил генерацию кода с последующим возвратом результата в разрабатываемый проект.

Результат эксперимента подтвердил жизнеспособность предложенного метода интеграции, так как задачи интеграции систем были решены без непосредственного вмешательства в исходные коды за счет использования средств конфигурирования динамического ПИ.

Литература

1. *Rich S.* Integration Imperative журнал Oracle Magazine, no.6, 2004, раздел "ТЕХНОЛОГИЯ", рубрика "Промышленный стандарт", <http://www.oracle.com/technology/oramag/oracle/04-nov/o64industry.html>
2. *Ierusalimschy R., Figueiredo L. H., Celes W.* The Evolution of an Extension Language: A History of Lua // Proc. of V Brazilian Symposium on Programming Languages, 2001. B-14–B-28.
3. *Гутцалов Н. В., Пауконен А. М., Скорлышев Д. Ю., Чубраев А. А. и др.* Среда разработки прикладных программ // Материалы конф. «Региональная информатика-2004». СПб, 2004. С. 65–69.
4. *Скорлышев Д. Ю., Ассонов А. А., Пауконен А. М., Чубраев А. А. и др.* Адаптивная инструментальная система для разработки программного обеспечения // Изв. вузов. Приборостроение. 2006. Т. 49, № 11. С. 65–69.

Скорлышев Дмитрий Юрьевич — мл. науч. Сотр. лаборатории технологий и систем программирования Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: инструментальные системы разработки ПО, технологии построения бизнес-приложений. Число научных публикаций — 2. Адрес: skorlyshev@rambler.su; СПИИРАН, 14-я линия В.О., д. 39, Санкт-Петербург, 199178, РФ; раб. тел. +7(812)328-3337.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Skorlyshev Dmitry Yurievich — junior researcher, Laboratory of Programming Technologies and Systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: instrumental systems of soft development, technologies of business applications. The number of publications — 2. Address: skorlyshev@rambler.ru; SPIIRAS, 39, 14th Line V. O., St. Petersburg, 199178, Russia; office phone +7(812)328-3337.

Шкиртиль Вячеслав Иванович — канд. техн. наук, заведующий лабораторией технологий и систем программирования Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: технологии программирования систем реального времени. Число научных публикаций — 32. Адрес: jvatlas@mail.rcom.ru; СПИИРАН, 14-я линия В. О., д. 39, Санкт-Петербург, 199178, РФ; раб. тел. +7(812)328-3337.

Shkirtil Viacheslav Ivanovich — PhD, chief of Laboratory of Programming Technologies and Systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: programming technologies of real time systems. The number of publications — 32. Address: jvatlas@mail.rcom.ru; SPIIRAS, 39, 14th Line V. O., St. Petersburg, 199178, Russia; office phone +7(812)328-3337.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru