

Goloso Peter Sergeevich – e-mail: golemstone@mail.ru; phone: +79518256497; the department of hydroacoustic and medical engineering; postgraduate student.

Orda-Zhigulina Dina Vladimirovna – e-mail: dinazhigulina@mail.ru; phone: +79081762662; the department of hydroacoustic and medical engineering; postgraduate student.

Starchenko Irina Borisovna – e-mail: star@sfnu.ru; phone: +78634371795; the department of hydroacoustic and medical engineering, dr. of eng. sc.; professor.

УДК 004.4'22

Ю.И. Рогозов, А.А. Дегтярев

**МЕТОД КОНФИГУРИРОВАНИЯ ФУНКЦИОНАЛЬНОСТИ
ПРОГРАММНЫХ СРЕДСТВ ГИДРОАКУСТИЧЕСКИХ
ИНФОРМАЦИОННЫХ СИСТЕМ***

Рассматривается проблема конфигурирования функциональности программных средств информационных систем. Исследуются достоинства и недостатки каркасного подхода, при котором основной акцент смещается с разработки уникальных систем на разработку семейства систем. Каркас – это по своей сути некоторая система, конфигурируя которую можно получать различные в ограниченном классе системы со статическими свойствами, т.е. осуществляющая некоторый процесс. В статье предлагается метод, предназначенный для конфигурирования системы под заданные требования, посредством использования механизма реализации действия, в котором элементы определенного класса являются динамическими, т.е. могут быть заменены или в случае необходимости сконфигурированы механизмом более низкого уровня. Для раскрытия механизма реализации действия вводятся специальные типы характеристик действия: методика, элементы, функции, инструменты, результаты. Также предлагается формальное представление механизма. Метод излагается в виде последовательности шагов. Приводятся сведения о реализации инструментария, поддерживающего метод, и о результатах его использования.

Конфигурирование; программные средства; гидроакустика.

Y.I. Rogozov, A.A. Degtyarev

**METHOD CONFIGURATING FUNCTIONAL OF SOFTWARE
FOR UNDERWATER ACOUSTICS INFORMATION SYSTEMS**

In this article is considering the problem of configuring functionality of information system software. In this case studying the advantages and disadvantages of the framework approach, in this main attention is shift from single system development to family systems development. Framework – this is something system, which can be obtained by configuring the different systems in a limited class of static properties, i.e. carrying out some process. A method intended for system configuration under given requirements, by using the mechanism of the action in which the elements of a particular class are dynamic, i.e. may be replaced, or if necessary configured by mechanism lower level. For decomposition low-level mechanism of action introduced new concept, such as technique, elements, function, tools, and results. Just invited to a formal representation mechanism. The method gives such as sequential of steps. It provides information on the implementation of tools that supporting the method, and the results of its use.

Configuring; software; underwater acoustics.

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 13-07-00971 а.

Анализ существующих подходов конфигурирования. Под конфигурированием программных средств понимается процесс приведения «компонента» или «системы» в соответствии с различными требованиями, без изменения основного кода [1]. В данной статье рассматривается конфигурирование функциональности программных средств. Это означает, что побуждающими необходимостью изменений являются новые функциональные требования.

За время существования программной инженерии появилось множество подходов [2], направленных на повышение эффективности разработки программных средств и, как следствие, возможности более эффективно приводить программные средства в соответствии с новыми требованиями к ним. Из всех известных подходов, непосредственно нацеленных на конфигурирование, на наш взгляд, можно считать каркасный подход, часто именуемый как framework [3].

Каркасный подход. Идея и недостатки. Основной идеей для каркасного подхода является создание неизменяемого программного ядра, с помощью которого можно создавать различные программные средства в рамках некоторой ограниченной предметной области. То есть акцент смещается с разработки уникальных систем на разработку семейства систем [4]. Для этого определяют наиболее общие задачи предметной области и выносят их в «статическую» часть, а задачи, которые от системы к системе меняются, выносят в «динамическую» часть, предоставляя при этом какие-либо механизмы обеспечения этой динамичности.

Известно, что каркасный подход имеет удачные практические реализации в различных предметных областях [5, 6]. Однако в большинстве реализаций механизмы конфигурирования реализованы как возможность замены или добавления модуля из некоторой библиотеки модулей. В случае когда необходимый модуль отсутствует, пользователю каркаса нужно «с нуля» программировать требуемый модуль, чаще всего на том языке программирования, на котором написан каркас. Это вызывает серьезные трудности, так как в большинстве своем пользователи каркаса не обладают достаточными знаниями в области программирования. Более того, на формальном и теоретическом плане внимание каркасному подходу практически не уделяется, что влечет отсутствие каких-либо руководств для разработки новых каркасов. В научных публикациях, например [7, 8], по смыслу пересекающихся с задачами конфигурирования программных средств под новые требования, основной акцент делается на языках и способах выражения новых требований. При этом упускается из вида то, что конфигурируемая единица является частью большей системы. Это по объективным причинам приводит к нарушению целостности, по крайней мере на формальном и теоретическом плане. Лишь в некоторых работах [9] авторы акцентируют внимание на выделении некоторых «статических» задач, характерных для рассматриваемого семейства систем, тем самым как бы подтверждая использование идей каркасного подхода.

Все вышеперечисленное позволяет полагать, что создание каркасов для новых предметных областей остается прерогативой искусства программирования и опыта системных аналитиков. Поэтому существует острая необходимость в теоретической проработке каркасного подхода. Далее в статье делается попытка такой проработки на примере метода конфигурирования функциональности программных средств гидроакустических информационных систем (ГА ИС) [10], функциональность которых определяется алгоритмами обработки данных, алгоритмами управления параметрами излучения и их связями друг с другом и другими элементами (аспектами) программного средства.

Метод конфигурирования функциональности. Каркас – это по своей сути некоторая система, конфигурируя которую можно получать различные в ограниченном классе системы со статическими свойствами, т.е. осуществляющая некоторый процесс. При такой точке зрения можно установить тождественное равенство

между понятием каркаса и понятием действия. Для любого действия можно раскрыть механизм его реализации. В то же время любой компонент механизма реализации действия, при необходимости его создания или изменения, может быть рассмотрен как действие более низкого уровня, что соответствует пониманию системы как действий, вложенных в действия [11].

Основная суть предлагаемого метода заключается в конфигурировании системы под заданные требования посредством использования механизма реализации действия, в котором элементы определенного класса являются динамическими, т.е. могут быть заменены или в случае необходимости сконфигурированы механизмом более низкого уровня. Конфигурирование при такой постановке заключается в выборе/конфигурировании необходимых динамических элементов и получении статической модели системы, в которой выбранные/сконфигурированные элементы жестко связаны с остальными статическими элементами механизма реализации действия.

Для раскрытия механизмов будем использовать типы характеристик, предложенные для области программных средств информационных систем [11]:

- ◆ *методика* – общие правила взаимодействия с элементами, функциями, инструментами для субъекта, выполняющего действие, обеспечивающие получение результатов (объектов) определенного класса;
- ◆ *элементы* – это «материя» или «сущности», объединение, соединение, преобразование или обработка, подмножества которых обеспечивают получение результата (объекта);
- ◆ *функции* – это такие операции, которые можно применять к одному или нескольким элементам, в результате выполнения функции могут появляться производные элементы;
- ◆ *инструменты* – это технологический базис, который позволяет физически получить доступ к элементам и выполнять над ними функции;
- ◆ *результаты (объекты, модели)* – это следствие деятельности индивида (субъекта) по одной из множества методик, с помощью подмножества множества инструментов, по выполнению подмножества множества функций над подмножеством множества элементов.

Таким образом, для того чтобы получить статическую модель программного средства ГА ИС, необходимо реализовать в каркасе предлагаемый метод, который состоит из следующей последовательности шагов:

Шаг 1. Определить класс гидроакустического устройства (одноканальное, двухканальное, трехканальное, пассивное, активное), в составе которого должно работать программное средство, функциональность которого конфигурируется.

Шаг 2. Выбрать методику из множества методик механизма реализации действия «конфигурирование функциональности программных средств ГА ИС», соответствующую классу, определенному на шаге 1.

Шаг 3. Сформировать список динамических элементов, необходимых для решения задачи и требования к ним.

Шаг 4. Найти элементы, которые позволят удовлетворить требования, сформированные на шаге 3.

Шаг 5. Если необходимые элементы присутствуют в списке доступных, то перейти к шагу 7, если отсутствуют – то к шагу 6.

Шаг 6*. Сформировать необходимые элементы, используя вложенный механизм, соответствующий классу требуемого элемента.

Шаг 7. Перейти к выполнению функций над выбранными (или сконфигурированными) элементами в соответствии с методикой из шага 1.

Шаг 8. В составе ГА ИС оценить полученную функциональность на предмет соответствия требованиям из шага 3. В случае необходимости повторить шаги 4–7.

*Примечание: Для вложенных механизмов метод идентичен представленному выше, за исключением того, что для них отсутствуют шаги 5 и 6.

В приведенной последовательности шаги 1–3 соответствуют этапу формирования требований к конфигурируемому программному средству, шаги 4–7 соответствуют этапу конфигурирования функциональности, а шаг 8 предназначен для проверки полученного программного средства на соответствие предъявляемым к нему требованиям.

Структура механизма реализации действия «конфигурирование функциональности программных средств ГА ИС», который является основной составляющей частью предлагаемого метода, приведена в табл. 1.

Таблица 1

Структура механизма реализации действия

Методики ($M^{El_3^{GAIS}}$)	Элементы $El_b^{El_3^{GAIS}}$		Функции $Func_c^{El_3^{GAIS}}$		Результаты $Result_e^{El_3^{GAIS}}$	
<p>Одноканальная гидроакустическая система - $m_1^{El_3^{GAIS}}$</p> <p>1. Загрузить и зарегистрировать плагин обработки</p> <p>2. Связать плагин из п. 1 с адаптером</p> <p>3. Повторить шаг 1 и 2 необходимое число раз;</p> <p>4. Загрузить и зарегистрировать плагин управления</p> <p>5. Связать плагин из п. 4 с адаптером</p>	<p>Адаптеры $El_1^{El_3^{GAIS}}$</p>	$el_1^{El_1^{El_3^{GAIS}}}$	<p>Функции класса времени подготовки системы к работе $Func_1^{El_3^{GAIS}}$</p>	Загрузить и зарегистрировать плагин $f_1^{Func_1^{El_3^{GAIS}}}$	<p>$Result_1^{El_3^{GAIS}}$ семейство результатов, соответствующих методике 1, т.е. одноканальные гидроакустические системы</p>	
		<p>Плагины обработки $El_2^{El_3^{GAIS}}$</p>				Связать плагин класса обработка с адаптером $f_2^{Func_1^{El_3^{GAIS}}}$
						Связать плагин класса управления с адаптером $f_3^{Func_1^{El_3^{GAIS}}}$
			Включить плагин $f_1^{Func_2^{El_3^{GAIS}}}$			
				Выключить плагин $f_2^{Func_2^{El_3^{GAIS}}}$		
	<p>Плагины управления $El_3^{El_3^{GAIS}}$</p>					
	<p>Делегаты $El_4^{El_3^{GAIS}}$</p>		$el_1^{El_4^{El_3^{GAIS}}}$			

В таблице не приведен тип характеристик «Инструменты», так как он представляет собой программную реализацию каркаса, поддерживающего рассматриваемый метод. Элементы подклассов «плагины обработки» и «плагины управления» являются динамическими элементами, в то время как элементы подклассов «Адаптеры» и «Делегаты» являются статическими.

Формально механизм реализации действия «конфигурирование функциональности программных средств ГА ИС» представляется совокупностью

$$Mehanism_{El_3^{GAIS}} = \langle M_{El_3^{GAIS}}, \{El_1^{El_3^{GAIS}}, El_2^{El_3^{GAIS}}, El_3^{El_3^{GAIS}}, El_4^{El_3^{GAIS}}\}, \{Func_1^{El_3^{GAIS}}, Func_2^{El_3^{GAIS}}\}, Tools_1^{El_3^{GAIS}}, Result_1^{El_3^{GAIS}} \rangle. \quad (1)$$

Механизм реализации действия для шага 6 из предлагаемого метода, который является вложенным в $Mehanism_{El_3^{GAIS}}$, представляется совокупностью

$$Mehanism_{El_{2,3}^{El_3^{GAIS}}} = \langle M_{El_{2,3}^{El_3^{GAIS}}}, El_b^{El_{2,3}^{El_3^{GAIS}}}, Func_c^{El_{2,3}^{El_3^{GAIS}}}, Tools_d^{El_{2,3}^{El_3^{GAIS}}}, Result_e^{El_{2,3}^{El_3^{GAIS}}} \rangle. \quad (2)$$

Тогда формально получение элемента, являющего элементом множества статичных элементов в механизме $Mehanism_{El_3^{GAIS}}$, записывается следующим образом:

$$el_n^{El_{2,3}^{El_3^{GAIS}}} = r_k^{Result_e^{El_{2,3}^{El_3^{GAIS}}}} = F(m_e^{M_{El_{2,3}^{El_3^{GAIS}}}}, \langle El_b^{El_{2,3}^{El_3^{GAIS}}}, Func_c^{El_{2,3}^{El_3^{GAIS}}}, Tools_d^{El_{2,3}^{El_3^{GAIS}}} \rangle), \quad (3)$$

где F – это последовательность императивных операций, выражаемых как выполнение функций из множества функций над элементами с помощью инструментов, которая выполняется субъектом, взаимодействующим с механизмом по выбранной методике. В области ответственности субъекта остается выбор конкретных элементов, над которыми будут выполняться функции, в то время как общая последовательность императивов определяется методикой.

Заключение. Основным достоинством предложенного метода можно считать вложенность механизмов. Формальное представление вложенности механизмов позволяет не только упростить процедуру конфигурирования программных средств, но и сохранять и отображать семантический характер элементов механизма (содержательный характер).

Программная реализация прилагаемого метода позволяет перейти к более систематическому повторному использованию кода и проектных и алгоритмических решений.

Спроектированный и реализованный инструментарий [12, 13], по предварительным оценкам, позволил сократить время разработки программного средства для промерного эхолота [14] на 35 %.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Хавьер Гарсия, Герман Голдшмидт. Разработка составных бизнес-сервисов на базе сервисно-ориентированной архитектуры. <<http://www.ibm.com/developerworks/ru/library/ws-soa-composite/>> (12.05.2013).
2. Рогозов Ю. И., Свиридов А. С., Дегтярев А. А. Анализ и перспективы развития разработки программного обеспечения // Информационные технологии. – 2011. – № 12. – С. 16-21.
3. Дегтярев А.А. Анализ способов сокращения семантического разрыва при разработке программного обеспечения // Технологии разработки информационных систем ТРИС-2012: материалы конференции. Т. 1. – Таганрог. Изд-во ТТИ ЮФУ, 2012. – С. 146-151.
4. Рогозов Ю. И., Свиридов А.С., Дегтярев А.А. Построение классификации характеристик программного обеспечения с целью идентификации понятий предметной области как характеристик // Информатизация и связь. – 2011. – № 3. – С. 80-83.
5. Magento™ creates huge success with enterprise e-commerce platform & community built on Zend Framework. <<http://www.docstoc.com/docs/44455580/Magento-%C3%A2%E2%80%9E%C2%A2-creates-huge-success-with-enterprise-e-commerce-platform---PDF>> (12.05.2013).
6. Microsoft Dynamic CRM. <<http://www.microsoft.com/rus/dynamics/crm/>> (12.05.2013).

7. *Лядова Л.Н.* Метамоделирование как основа средств оперативной разработки профессионально-ориентированных информационных систем // Математика программных систем: межвузовский сборник научных статей. – Пермь: Пермский государственный национальный исследовательский университет, 2012. – С. 20-32.
8. *Athula Ginige.* Meta-design paradigm based approach for iterative rapid development of enterprise WEB applications // Proceedings of the Fifth International Conference on Software and Data Technologies, ICSOFT 2010. – P. 337-343.
9. *Youri I. Rogozov, Alexander S. Sviridov, Sergey A. Kutcherov, Wladimir Bodrov.* Purpose-driven approach for flexible structure-independent database design. Proceedings of the Fifth International Conference on Software and Data Technologies, ICSOFT 2010. – Vol. 1. – P. 356-362.
10. *Рогозов Ю.И., Дегтярев А.А.* Информационные процессы в малоканальных гидроакустических комплексах ближнего действия, построенных на основе программно-ориентированной архитектуры // ИТСАУ11 IX Всероссийская научная конференция молодых учёных, аспирантов и студентов «Информационные технологии, системный анализ и управление»: Сборник материалов. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – Т. 1. – С. 64-68.
11. *Рогозов Ю.И.* Понятие метасистемы как методологической основы создания системы // Промышленные АСУ и контроллеры. – 2013. – № 2. – С. 38-46.
12. *Дегтярев А.А.* Концепция инструментального средства для построения и адаптации информационных систем // Информационные технологии, системный анализ и управление – ИТСАиУ-2012: Сборник трудов X Всероссийской научной конференции молодых ученых аспирантов и студентов. – Таганрог: Изд-во ЮФУ, 2012. – Т. 1. – С. 43-46.
13. *Дегтярев А.А.* Архитектура программного ядра для построения информационных систем сбора, обработки и хранения данных на примере малоканальных гидроакустических систем // Высокопроизводительные вычислительные системы: Сб. науч. тр. Вып. 2. – Таганрог: Изд-во ЮФУ, 2012. – С. 7-11.
14. *Заковоротнов Е.А., Дегтярева Е.Е., Дегтярев А.А., Малыхина О.В.* Программа оперативного сбора, обработки и хранения информации для промерного эхолота // Свидетельство о государственной регистрации программы для ЭВМ №20111618931 от 16 ноября 2011 г.

Статью рекомендовал к опубликованию к.т.н. Е.А. Заковоротнов.

Рогозов Юрий Иванович – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: rogozov@tti.sfedu.ru; 347928, г. Таганрог, Некрасовский, 44; тел.: 88634371787; кафедра системного анализа и телекоммуникаций; зав; кафедрой; д.т.н.; профессор.

Дегтярев Алексей Андреевич – e-mail: alexey.a.degtyarev@gmail.com; кафедра системного анализа и телекоммуникаций; аспирант.

Rogozov Yuri Ivanovich – Federal State-owner Autonomy Educational Establishment of Higher Vocational Educational “Southern Federal University”; e-mail: rogozov@tti.sfedu.ru; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +78634371787; the department of system analysis and telecommunications; head the department; dr. of eng. sc.; professor.

Degtyarev Alexey Andreevich – e-mail: alexey.a.degtyarev@gmail.com; the department of system analysis and telecommunications; postgraduate student.