

Інфокомунікаційні системи

УДК 004.03;

М.В. Евланов, В.А. Никитюк

Харьковский национальный университет радиоэлектроники, Харьков

ОБОБЩЕННАЯ МОДЕЛЬ ФРЕЙМВОРКА ФУНКЦИОНАЛЬНОГО СЕРВИСА ИНФОРМАЦИОННОЙ СИСТЕМЫ

Рассматривается задача разработки модели программного обеспечения функционального сервиса как элемента сервис-ориентированной информационной системы. В качестве основного шаблона разработки программного обеспечения функционального сервиса выбран паттерн MVC (Model–View–Controller). Предложена обобщенная категорная модель фреймворка функционального сервиса информационной системы. Модель описывает основные связи между представлениями различных компонентов фреймворка и семантического описания функционального сервиса.

Ключевые слова: сервис, паттерн проектирования, Model–View–Controller, фреймворк, семантическое представление.

Введение

Проблемы внедрения и эксплуатации современных информационных систем. Внедрение и эксплуатация различных информационных систем (ИС) и технологий (ИТ) на крупных и средних предприятиях и организациях как объектах управления (ОУ) привели к появлению новых, ранее не возникавших проблем. Одной из таких проблем является формирование в процессе эксплуатации различных ИС больших массивов разнородных данных, которые требуют надежного хранения, быстрой и согласованной обработки. Предполагается, что на основе этих данных руководители должны принимать управленческие решения, связанные с расширением и стабилизацией бизнес-процессов ОУ.

Под термином «бизнес-процесс» здесь и далее (согласно стандарту ISO 9000:2005) следует понимать совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующих входы (материальные, информационные потоки и т.п.) в выходы, представляющие ценность для потребителя.

Проблема организации быстрой согласованной обработки данных из массивов информации, формируемых различными ИС, путем интеграции отдельных ИС и информационных технологий является одной из ключевых проблем построения единой системы управления современным предприятием или организацией.

Эта проблема не имеет единого теоретического решения, а большинство работ в данной области посвящено разработке концептуальных решений, не имеющих формализованного описания и, следовательно, не позволяющих объективно оценить эффективность интеграции разнородных ИС в единую систему управления предприятием.

Анализ сервис-ориентированной архитектуры как подхода к решению проблемы интеграции разнородных информационных систем.

В соответствии с последними тенденциями в развитии ИТ основой для построения ИС корпоративного масштаба служит сервис-ориентированная архитектура (Service-Oriented Architecture, SOA), которая предполагает построение ИС из набора гетерогенных слабосвязанных компонентов (сервисов) [1].

SOA понимается как парадигма организации и использования распределенного множества функций, которые могут контролироваться различными владельцами. В основу данной архитектуры положено понятие «информационный сервис». Информационный сервис – это атомарная прикладная функция автоматизированной системы, которая пригодна для использования при разработке приложений, реализующих прикладную логику автоматизируемых процессов как в самой системе, так и для использования в приложениях других автоматизированных систем.

Сервис обычно характеризуется следующими свойствами:

- возможностью многократного применения;
- услуга может быть определена одним или несколькими технологически независимыми интерфейсами;
- выделенные услуги слабо связаны между собой, и каждая из них может быть вызвана посредством коммуникационных протоколов, обеспечивающих возможность взаимодействия услуг между собой.

Обязательным условием построения и внедрения архитектуры системы на основе SOA является использование единой инфраструктуры описания сервисов (репозитория сервисов), разрешенных протоколов доступа и обмена сообщениями, форматов сообщений.

Упомянутая инфраструктура образует так называемую интеграционную шину (ИШ) (Enterprise Service Bus – ESB), являющуюся одним из центральных компонентов системы. Она устанавливает единые правила публикации сервисов, управления и информационного взаимодействия между приложениями различных систем, входящих в состав интегрированной системы. Это упрощает управление приложениями и их поддержку, а также снижает риск фрагментации приложений и процессов [1].

Каждая из служб взаимодействует не с остальными службами напрямую, а только с шиной. ИШ образует однородную среду информационного взаимодействия и является фундаментом для интеграции информационных систем, функционирующих в различных учреждениях и ведомствах. ИШ определяет, кем, где, каким образом и в каком порядке должны обрабатываться запросы.

Выделение нерешенной части проблемы. Рассмотренные особенности SOA достаточно просты с точки зрения теории построения систем. Однако на практике усилия на разработку и внедрение корневых компонентов SOA, собственно и образующих ИШ, – репозитория и хранилища SOA – во многих случаях затрачиваются не настолько правильно, чтобы SOA-ИС могла функционировать успешно. Более того, руководство предприятия склонно забывать о необходимости эффективного управления данными и ИТ-сервисами до тех пор, пока не становится слишком поздно [2]. Такое опоздание приводит к неоправданным затратам финансовых и других ресурсов на эксплуатацию отдельных сервисов ИС предприятия без возможности окупить эти затраты за счет эффекта от эксплуатации ИС предприятия в целом.

Другой, не менее важной проблемой информатизации предприятий, является уже отмеченное выше разнообразие поставщиков и решений на рынке SOA-ИС. Такое разнообразие приводит к тому, что SOA-ИС целого ряда предприятий формируются из разнородных ИТ-сервисов. Вследствие этого возникает интерес к решению проблемы повышения эффективности использования ИТ в основной деятельности предприятия и к оптимизации затрат расходовемых при этом ресурсов различного рода. Эту задачу не следует считать элементарной или же типовой. Есть большое количество примеров того, как работы по информатизации предприятия не дают желаемого эффекта или же приводят к излишним трудовым затратам [10].

Сказанное позволяет считать задачу управления разнородными ИТ-сервисами в значительной степени нерешенной.

Постановка задачи исследования

Для решения задачи управления разнородными ИТ-сервисами необходимо, прежде всего, разрабо-

тать комплекс моделей ИТ-сервиса как основного элемента ИС управления предприятиями и организациями. В качестве основы для построения подобного комплекса моделей следует использовать рассмотренные особенности SOA как одного из наиболее перспективных подходов к интеграции отдельных элементов ИС управления предприятиями и организациями. При этом основное внимание следует уделять разработке моделей программного обеспечения (ПО) ИТ-сервисов как вида обеспечения ИС, наиболее часто изменяющегося в процессе эксплуатации ИС.

Обоснование выбора шаблона проектирования программного обеспечения функционального сервиса.

Как сказано выше, сервисы с точки зрения SOA рассматриваются как гетерогенные слабосвязанные элементы ИС. Такое представление означает, что связи между отдельными компонентами сервисов сведены к некоему минимуму, который определяется выбранным разработчиками шаблоном (паттерном) разработки отдельных видов обеспечений ИС и, прежде всего, программного обеспечения ИС.

Одним из наиболее распространенных шаблонов разработки ПО ИС является шаблон проектирования «Model – View – Controller» (MVC), название которого можно перевести как «Модель – вид – контроллер» или «Модель – представление – контроллер» [3]. Этот шаблон состоит из трех основных компонентов:

- компонент «модель» (model), который предоставляет знания, данные и методы для работы с этими данными, реагирует на запросы, изменяя при этом свое состояние, но не содержит знаний и способов визуализации знаний и данных;

- компонент «представление» или «вид» (view), который обеспечивает визуализацию данных в той или иной форме (окнах) с использованием графических элементов;

- компонент «контроллер» (controller), который контролирует ввод данных пользователем и использует остальные компоненты для реализации необходимой реакции ПО ИС на действия пользователя.

В настоящее время выделяют две основные разновидности MVC: пассивная модель и активная модель. Пассивная модель не имеет никаких способов и методов воздействия на другие компоненты MVC и используется этими компонентами в качестве источника данных для отображения. Все изменения пассивной модели отслеживаются контроллером, который также обеспечивает соответствующие изменения представления. Активная модель рассылает представлениям ПО ИС сообщения о происходящих в модели изменениях (в том числе, выполнении тех или иных методов обработки данных), а представления, заинтересованные в получении тех

или иных сообщений об изменениях, подписываются на соответствующую рассылку. Контроллер при этом выполняет ограниченный набор типовых команд управления [4]. Такая разновидность MVC наиболее пригодна для создания гибких ИС, элементы которых должны быть максимально независимы друг от друга, а бизнес-логика (алгоритмы решения функциональных задач ИС) должна выполняться не на компьютерах пользователей, а на сервере. К таким ИС относятся многочисленные системы, предназначенные для работы в Интернет/Инtranет сетях, а также ряд ИС, использующих распределенные базы данных (БД).

Обобщенное представление взаимосвязей компонентов шаблона MVC показано на рис. 1 [4]. Связи, характерные только для активной модели шаблона MVC, показаны на рисунке пунктиром.

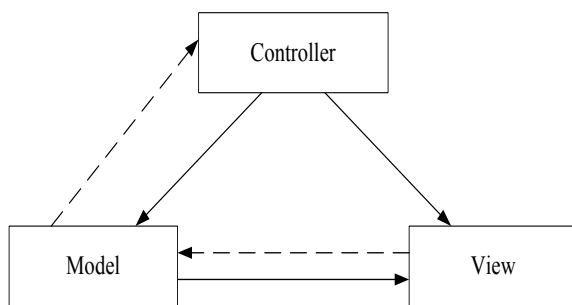


Рис. 1. Схема взаимосвязей компонентов шаблона проектирования «Model – View - Controller»

Одной из главных особенностей ИС, основанных на SOA, является гибкость, а использование ИШ предполагает не только возможность, но и целесообразность распределенной организации БД таких ИС. Поэтому использование разновидности «активная модель» шаблона проектирования MVC является наиболее целесообразным для разработки сервисов, обеспечивающих реализацию функций ИС, основанных на SOA. Такие сервисы предлагается здесь и в дальнейшем называть «функциональные сервисы».

Основываясь на приведенных выше определениях сервисов, функциональный сервис следует представлять как совокупность компонентов различных видов обеспечений, реализующих законченную функциональную задачу (или отдельную функцию задачи) ИС путем трансформации множества входных данных через последовательность целостных состояний обрабатываемых данных в множество выходных данных с использованием для межкомпонентного обмена данными стандартных платформо-независимых интерфейсов. Основное назначение таких интерфейсов – стандартизация операций распространения данных и знаний между различными сервисами и их компонентами в процессе эксплуатации ИС, основанной на SOA.

Данное определение позволяет на основе использования шаблона проектирования MVC сформировать модель фреймворка функционального сервиса ИС, основанной на SOA. Под термином «фреймворк» (или «каркас») следует понимать ПО, облегчающее разработку и объединение разных компонентов большого программного проекта. Часто термин «фреймворк» употребляют, говоря о постоянной части прикладного ПО, которая не меняется при смене версий ПО и несет в себе гнезда для сменных программных модулей или точек расширения прикладного ПО [5 – 8]. Применительно к функциональным сервисам ИС, основанной на SOA, термин «фреймворк» следует трактовать как постоянную часть (каркас) прикладного ПО, которая определяет границы отдельных компонентов шаблона проектирования ПО функционального сервиса, а также способы и правила взаимодействия этих компонентов в соответствии с выбранным шаблоном проектирования.

Разработка обобщенной модели фреймворка функционального сервиса информационной системы

В общем случае фреймворк функционального сервиса может рассматриваться как совокупность трех основных компонентов шаблона проектирования MVC и типовых правил обмена данными, командами и сообщениями между этими компонентами. Каждый из этих компонентов может рассматриваться как последовательность состояний, обладающих определенной структурой. Поэтому наиболее целесообразным для формализованного описания модели фреймворка следует признать категорно-функторный математический аппарат. Этот аппарат обладает следующими достоинствами:

- возможность описания программной реализации каждого компонента шаблона проектирования MVC в виде категории структурированных множеств классов, атрибутов и методов;
- возможность описания взаимодействий отдельных компонентов шаблона проектирования MVC как функторных отображений категорий друг в друга;
- возможность представления функторов, отображающих категорные описания компонентов шаблона проектирования MVC друг в друга, как набора правил или методов, реализующих эти отображения независимо от конкретного варианта реализации категорных описаний компонентов.

Обобщенная теоретико-категорная модель фреймворка функционального сервиса, основанного на шаблоне проектирования MVC, будет иметь следующий вид:

$$L_{ITS} = [L_M, L_{Int}, L_C, F_{L_M}^{L_C}, F_{L_{Int}}^{L_C}, F_{L_M}^{L_{Int}}, F_{L_{Int}}^{L_M}, F_{L_C}^{L_{Int}}], (1)$$

где L_{ITS} – обозначение теоретико-категорной модели фреймворка функционального сервиса; L_M – подкатегория, описывающая компонент «модель» фреймворка функционального сервиса; L_{Int} – подкатегория, описывающая компонент «представление» фреймворка функционального сервиса; L_C – подкатегория, описывающая компонент «контроллер» фреймворка функционального сервиса; $F_{L_M}^{L_C}$ – функтор, описывающий отображение категорного описания компонента «контроллер» в категорное описание компонента «модель»; $F_{L_{Int}}^{L_C}$ – функтор, описывающий отображение категорного описания компонента «контроллер» в категорное описание компонента «представление»; $F_{L_M}^{L_{Int}}$ – функтор, описывающий отображение категорного описания компонента «представление» в категорное описание компонента «модель»; $F_{L_{Int}}^{L_M}$ – функтор, описывающий отображение категорного описания компонента «модель» в категорное описание компонента «представление»; $F_{L_C}^{L_{Int}}$ – функтор, описывающий отображение категорного описания компонента «представление» в категорное описание компонента «контроллер».

Функторы $F_{L_M}^{L_C}$ и $F_{L_{Int}}^{L_C}$ описывают методы и способы преобразования наборов команд, сформированных компонентом «контроллер», в наборы команд, исполняемых элементами компонентов «модель» и «представление» соответственно. Функтор $F_{L_M}^{L_{Int}}$ описывает методы и способы организации данных, которые передаются из компонента «представление» в компонент «модель» в ходе работы фреймворка функционального сервиса. Функтор $F_{L_{Int}}^{L_M}$ описывает методы и способы организации описаний событий, возникающих в ходе работы компонента «модель», которые влияют на работу компонента «представление». Функтор $F_{L_C}^{L_{Int}}$ описывает методы и способы организации описаний событий, возникающих в ходе работы компонента «представление», которые влияют на работу компонента «контроллер».

Данная модель позволяет описать варианты фреймворков функционального сервиса, основанные как на пассивной, так и на активной модели шаблона проектирования MVC. В случае пассивной модели шаблона функторы $F_{L_{Int}}^{L_M}$ и $F_{L_C}^{L_{Int}}$ могут быть не определены, а соответствующие методы и способы

не использоваться в ходе работы фреймворка.

Однако данная модель не позволяет описать процедуры настройки функционального сервиса на особенности функционирования в ИС, основанной на SOA. Так, например, выше было отмечено, что в такой системе компонент «модель» каждого функционального сервиса s_i должен рассылать сообщения с результатами своей работы. Эти сообщения могут принять и использовать любые другие функциональные сервисы, использующие результаты работы функционального сервиса s_i . Таким образом, модель (1) необходимо уточнить с учетом следующих особенностей фреймворка функционального сервиса:

- особенность отображения знаний о внутренней структуре и содержании каждого компонента фреймворка в описания соответствующих компонентов;

- особенность внутреннего строения компонента «модель» для разновидности «активная модель» шаблона проектирования MVC.

Первая из отмеченных особенностей позволит рассматривать функциональный сервис с точки зрения знаний о его структуре и содержании. Реализация этой особенности позволит создать в рамках фреймворка новый механизм управления структурой и содержанием отдельных компонентов путем формирования и изменения знаний (в частности, метаданных) о функциональном сервисе. Такой механизм значительно облегчит процесс интеграции отдельных сервисов в множество сервисов, эксплуатируемых в рамках ИС, основанной на SOA.

Реализация подобного механизма невозможна без уточнения описания компонента «модель» разновидности «активная модель» шаблона проектирования MVC. Как отмечалось выше, этот компонент является наиболее сложным и отвечает за реализацию обработки данных в ходе решения функциональной задачи ИС, основанной на SOA, на самых различных уровнях (уровне хранения данных, уровне обработки потоков данных, уровне бизнес-логики).

С учетом данных особенностей схему взаимосвязей отдельных компонентов фреймворка функционального сервиса, основанного на разновидности «активная модель» шаблона проектирования MVC, можно представить следующим образом (см. рис. 2).

Тогда категорное описание компонента «модель» L_M с учетом предлагаемого механизма управления структурой и содержанием отдельных компонентов фреймворка будет иметь следующий вид [9]:

$$L_M = [L_{Sem}, L_{DB}, L_{DF}, L_{BL}, F_{L_{DB}}^{L_{Sem}}, F_{L_{DF}}^{L_{DB}}, F_{L_{BL}}^{L_{DF}}, F_{L_{DB}}^{L_{DF}}, F_{L_{Sem}}^{L_{DB}}], \quad (2)$$

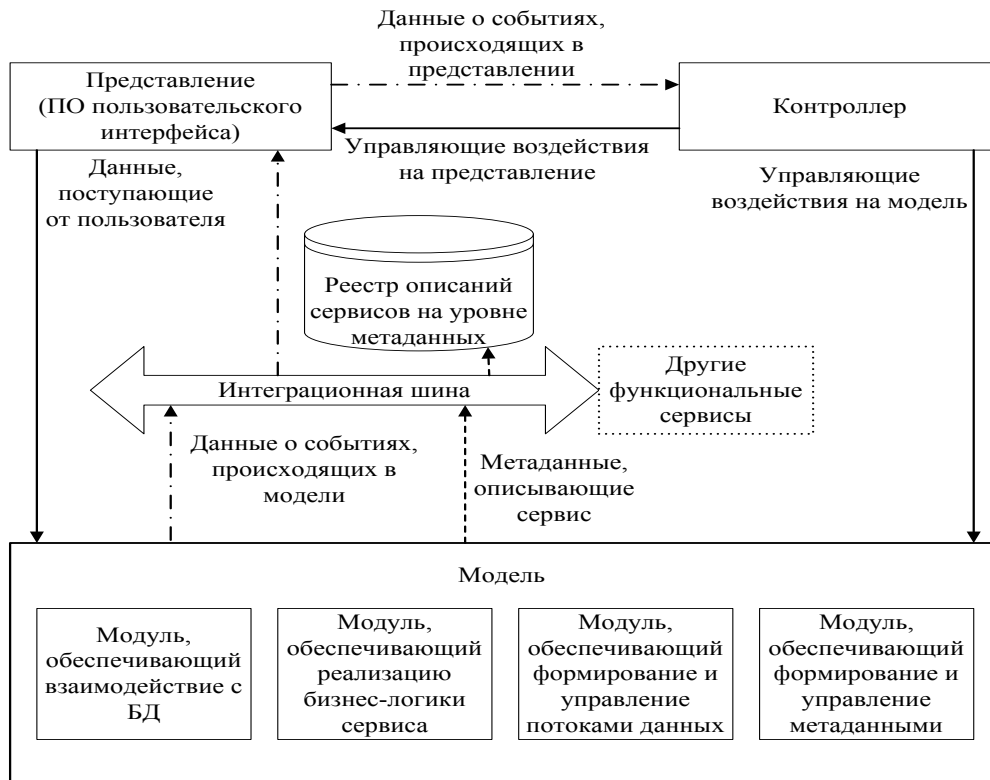


Рис. 2. Схема взаимосвязей основных компонентов фреймворка функционального сервиса

где L_{Sem} – обозначение семантического представления функционального сервиса как совокупности знаний (метаданных) о структуре и содержании отдельных компонентов этого сервиса; L_{DB} – обозначение представления функционального сервиса на уровне хранимых данных; L_{DF} – обозначение представления функционального сервиса на уровне потоков данных; L_{BL} – обозначение представления функционального сервиса на уровне программной реализации бизнес-логики; $F_{L_{DB}}^{L_{Sem}}$ – функтор, устанавливающий отображение семантического представления функционального сервиса в представление на уровне хранимых данных; $F_{L_{DF}}^{L_{DB}}$ – функтор, устанавливающий отображение представления функционального сервиса на уровне хранимых данных в представление на уровне потоков данных; $F_{L_{BL}}^{L_{DF}}$ – функтор, устанавливающий отображение представления функционального сервиса на уровне потоков данных в представление на уровне программной реализации бизнес-логики; $F_{L_{DF}}^{L_{BL}}$ – функтор, устанавливающий отображение представления функционального сервиса на уровне программной реализации бизнес-логики в представление на уровне потоков данных; $F_{L_{DB}}^{L_{DF}}$ – функтор, устанавливающий отображение представления функцио-

нального сервиса на уровне потоков данных в представление на уровне хранимых данных; $F_{L_{Sem}}^{L_{DB}}$ – функтор, устанавливающий отображение представления функционального сервиса на уровне хранимых данных в семантическое представление.

С учетом уточнения описания компонента «модель» в виде выражения (2) и предлагаемого к реализации механизма управления структурой и содержанием отдельных компонентов путем формирования и изменения знаний (в частности, метаданных) о функциональном сервисе, модель (1) примет следующий вид:

$$L_{ITS} = [L_{Sem}, L_{DB}, L_{DF}, L_{BL}, L_{Int}, L_C, F_{L_{DB}}^{L_{Sem}}, F_{L_{DF}}^{L_{DB}}, F_{L_{BL}}^{L_{DF}}, F_{L_{Int}}^{L_{BL}}, F_{L_{BL}}^{L_{Int}}, F_{L_{DF}}^{L_{BL}}, F_{L_{DB}}^{L_{DF}}, F_{L_{Sem}}^{L_{DB}}, F_{L_{Sem}}^{L_C}, F_{L_{DB}}^{L_C}, F_{L_{DF}}^{L_C}, F_{L_{BL}}^{L_C}, F_{L_{Int}}^{L_C}, F_{L_C}^{L_{Int}}],$$

где $F_{L_{Int}}^{L_{BL}}$ – функтор, устанавливающий отображение представления функционального сервиса на уровне программной реализации бизнес-логики в категорное описание компонента «представление»; $F_{L_{Int}}^{L_{BL}}$ – функтор, устанавливающий отображение категорного описания компонента «представление» функционального сервиса в представление на уровне программной реализации бизнес-логики; $F_{L_{Sem}}^{L_C}$ – функтор, описывающий отображение категорного описания компонента «контроллер» в се-

мантическое представление функционального сервиса; F_{LDB}^{LC} – функтор, описывающий отображение категорного описания компонента «контроллер» в представление функционального сервиса на уровне хранимых данных; F_{LDF}^{LC} – функтор, описывающий отображение категорного описания компонента «контроллер» в представление функционального сервиса на уровне потоков данных; F_{LBL}^{LC} – функтор, описывающий отображение категорного описания компонента «контроллер» в представление функционального сервиса на уровне программной реализации бизнес-логики.

Выводы и перспективы дальнейших исследований

Предлагаемая модель фреймворка функционального сервиса позволяет рассматривать подобные сервисы как совокупность описаний отдельных компонентов, связанных между собой типовыми преобразованиями данных и метаданных, и позволяющих изменять состав и содержание отдельных компонентов в зависимости от изменения знаний (метаданных) о структуре и содержании функционального сервиса. Подобный подход к формализованному представлению функционального сервиса позволяет представить проблему интеграции разнородных функциональных сервисов как задачу оптимизации семантических представлений актуального множества функциональных сервисов по результатам выполнения типовых операций интеграции.

Рассмотренные в данной статье результаты позволяют выделить следующие основные перспективы дальнейших исследований: разработка детализированных описаний семантического представления функционального сервиса; разработка механизмов, реализующих функторы, связывающие семантическое представление функционального сервиса и представление сервиса на уровне хранимых данных;

разработка инструментальных средств, позволяющих осуществлять интеграцию разнородных функциональных сервисов в рамках единой целостной ИС управления предприятием или организацией.

Список литературы

1. Деревянко А.С. Технологии и средства консолидации информации: Учебное пособие [Текст] / А.С. Деревянко, М.Н. Соловук. – Х.: НТУ "ХПИ", 2008. – 432 с.
2. Parikh Ash. SOA в реальности [Электронный ресурс] / Ash Parikh, Murty Gurajada. – Сайт ERPNews. – Режим доступа к сайту: <http://erpnews.ru/doc2610.html>. – Заголовок с экрана.
3. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Издательский дом «Питер», 2006. – 368 с.
4. Фримен Эр. Паттерны проектирования [Текст] / Эр. Фримен, Эл. Фримен, К. Сьерра, Б. Бейтс. – СПб.: Издательский дом «Питер», 2011. – 650 с.
5. Тернер М.С.В. Основы Microsoft Solution Framework [Текст] / М.С.В. Тернер. – СПб.: Русская редакция, ИД «Питер», 2008. – 336 с.
6. Yuan M.J. Seam Framework: Experience the Evolution of Java EE [Текст] / M.J. Yuan, J. Orshalick, T. Heute. – N.Y.: Prentice Hall Ptr, 2009. – 504 p.
7. Фримен А. ASP.NET MVC 3 Framework с примерами на С# для профессионалов [Текст] / А. Фримен, С. Сандерсон. – СПб.: Вильямс, 2011. – 672 с.
8. Васвани В. Zend Framework: разработка веб-приложений на PHP [Текст] / В. Васвани. – СПб.: ИД «Питер», 2012. – 432 с.
9. Евланов М.В. Теоретико-категорная модель IT-сервиса информационной системы управления предприятием [Текст] / М.В. Евланов, В.А. Никитюк, О.И. Калмыкова // VI-я Международная научно-практическая конференция «Наука и социальные проблемы общества: информатизация и информационные технологии». Сборник научных трудов. – Х.: ХНУРЭ, 2011. – С. 321-322.
10. Бизнес и информационные технологии [Электронный ресурс] / Сайт INTUIT. – Режим доступа к ресурсу: <http://www.intuit.ru/department/itmngi/entarc/1/>. – Заголовок с экрана.

Поступила в редколлегию 10.12.2012

Рецензент: д-р техн. наук, проф. В.П. Пуятин, Харьковский национальный технический университет сельского хозяйства им. Петра Василенко, Харьков.

УЗАГАЛЬНЕНА МОДЕЛЬ ФРЕЙМВОРКА ФУНКЦІОНАЛЬНОГО СЕРВІСУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

М.В. Євланов, В.А. Никитюк

Розглядається задача розробки моделі програмного забезпечення функціонального сервісу як елемента сервіс-орієнтованої інформаційної системи. Як основний шаблон розробки програмного забезпечення функціонального сервісу обраний паттерн MVC (Model–View–Controller). Запропонована узагальнена категорна модель фреймворка функціонального сервісу інформаційної системи. Модель описує основні зв'язки між представленнями різних компонентів фреймворка й семантичного опису функціонального сервісу.

Ключові слова: сервіс, паттерн проектування, Model–View–Controller, фреймворк, семантичне представлення.

GENERALIZED MODEL OF INFORMATION SYSTEM FUNCTIONAL SERVICE FRAMEWORK

M.V. Evlanov, V.A. Nikityuk

The problem of developing a functional model of software services as part of a service-oriented information system is considered. As a basic design pattern of software functional service selected pattern MVC (Model–View–Controller). A generalized model of the categorical framework of functional service information system is offered. The model describes the basic relationship between representations of the various components of the framework and the semantic description of functional service.

Keywords: service, design pattern, Model–View–Controller, a framework, the semantic representation.