

УДК 004.75:681.3

ИСПОЛЬЗОВАНИЕ ДОКУМЕНТО-ОРИЕНТИРОВАННОГО ИНТЕФЕЙСА ПРИ ПОСТРОЕНИИ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Бутенко М.Л.

Восточноукраинский национальный университет им. Владимира Даля

Проводится анализ преимуществ и недостатков документо-ориентированного интерфейса при построении информационно-управляющих систем. Основное внимание уделяется особенностям построения программной архитектуры документо-ориентированного интерфейса. Проводится сравнительный анализ работы функционально-ориентированных и документо-ориентированных систем.

Введение

Теория и практика создания программных систем широко представлена в научных исследованиях и практических разработках, руководствах языков программирования. Архитектор программной системы имеет громадный выбор подходов и средств реализации. Однако, рынок программных продуктов свидетельствует об избирательности покупателей, востребованности конкретных разработок.

Данная публикация посвящена ключевым особенностям использования документо-ориентированного интерфейса при построении программных систем. Прежде всего речь идет о сложных информационно-управляющих системах, например, системах комплексной автоматизации бизнес-процессов предприятий, производственных фирм, бюджетных организаций или крупных образовательных организаций, таких как вузы. Характерными чертами таких программных систем являются: очень длинный цикл разработки и внедрения системы, постоянно меняющаяся конфигурация внутренних и внешних информационных потоков, большое количество пользователей, наличие нескольких хранилищ данных и т.п.

Цель публикации – описание преимуществ документо-ориентированных интерфейсов при разработке информационно-управляющих систем и разработка способов взаимодействия независимых документо-ориентированных интерфейсов.

1. Документо-ориентированный интерфейс

Термин «документо-ориентированный интерфейс» далеко не новый. Каждый человек, который работает за компьютером, создает, открывает и редактирует документы. Подробно останавливаться на особенностях взаимодействия пользователя и программной системы не имеет смысла. Более интересна реализация программных интерфейсов. В данном случае стоит рассмотреть две классические реализации подобных интерфейсов. Это программные интерфейсы фирмы Microsoft, главным образом Microsoft Office API [1] и Cocoa Document Architecture фирмы Apple [2]. Наибольший интерес представляет Cocoa Document Architecture, так как она направлена на общие принципы программной архитектуры, а не на взаимодействие с конкретным программным продуктом.

В настоящее время, документо-ориентированные интерфейсы применяются главным образом только в тех программных системах, где есть четкое понятие документа как файла (вне зависимости от того, где он физически расположен: локально, в Internet или, например, в почтовой программе). В остальных случаях используются функционально-ориентированные интерфейсы.

Бизнес-приложения можно разделить на несколько видов. Рассмотрим функционально-ориентированные, как самые распространенные, и документо-ориентированные. Традиционный подход использует функционально-ориентированные модели, где рабочий процесс отображается на функциональности приложения. Одна функция может оперировать с большим количеством всевозможных данных, полученных из разных источников.

Документо-ориентированный подход отличается от функционально-ориентированного тем, что вместо того, чтобы описывать функции, он руководствуется различными бизнес-объектами. Эти бизнес-объекты, как раз и отображаются в документах, хотя файла документа может и не быть. В качестве примера можно привести медицинскую карточку в поликлинике, данные по кредиту в банке или формирование учебного плана в вузе. Конечно же, подобные документы связаны с другими документами в контексте информационно-управляющей системы. Также такие документы могут быть динамическими, то есть в следующий момент времени они могут иметь уже другое содержание.

2. Архитектура информационно-управляющей системы

Документо-ориентированные системы часто имеют схожую архитектуру, имеющую документ как основной объект. Все типы документов являются потомками базового интерфейса. Базовый интерфейс содержит набор свойств и методов, которые должны быть реализованы в дочерних типах документов в обязательном порядке. Система имеет модульную организацию, в которой каждый модуль реализует один или несколько типов документов. Программный модуль полностью отвечает за свой тип документа. В идеальном случае родительская система не реализует какой-либо тип документа и умеет обращаться ко всем документам только по методам и свойствам базового интерфейса. Родительская система также отвечает за взаимодействие документов и реализует систему событий.

В качестве примера приведем список методов и свойств базового интерфейса: создание, просмотр, редактирование, печать, просмотр перед печатью, настройка печати (печать документа реализует программный модуль за него отвечающий), импорт, экспорт (используется несколько форматов импорта и экспорта), сохранение, закрытие, поиск и другие.

Проведем сравнительный анализ особенностей реализации интерфейсов функционально-ориентированных и документо-ориентированных систем (табл. 1).

3. Способы взаимодействия документо-ориентированных интерфейсов

Информационно-управляющая система может использовать только стандартные интерфейсы. В этом случае разработка всех документо-ориентированных интерфейсов

Таблица 1. Реализация интерфейсов функционально-ориентированных и документо-ориентированных систем

Действие	Функционально-ориентированная система (ФОС)	Документо-ориентированная система (ДОС)	Комментарий
Общий интерфейс пользователя	Сложный интерфейс всей системы. Чаще всего требуется дополнительное обучение.	Стандартный интерфейс главного приложения. Интерфейс конкретного типа документа зависит от его реализации. Чаще всего обучение не требуется.	Общий интерфейс пользователя значительно лучше в ДОС.
Создание бизнес-процесса	Описание реализуется в одном месте.	Для описания бизнес-процесса, возможно потребуются создание нескольких связанных документов. Возможно использование нескольких экранных форм.	Реализация значительно проще в ФОС. Удобство использования проще в ФОС.
Редактирование бизнес-процесса	Редактирование реализуется в одном месте, чаще всего используется та же форма, что и для создания.	Для редактирования может использоваться другая форма – форма редактирования документа.	Реализация значительно проще в ФОС. При изменении одного документа – удобство редактирования лучше в ДОС.
Удаление бизнес-процесса	Бизнес-процессы чаще всего удаляются по отдельности.	Возможно множественное удаление.	Удаление незначительно лучше в ДОС.
Взаимодействие бизнес-процессов	Выполняется из формы редактирования. Задается взаимодействие в четко ограниченных рамках для данной версии приложения.	Выполняется отдельно для нескольких документов. Возможно создание дополнительных документов.	Реализация значительно проще в ФОС. Удобство использования лучше в ДОС.
Классификатор бизнес-процессов	Является частью приложения	Является отдельным документом. Может быть несколько классификаторов.	Реализация проще в ФОС. Удобство использования лучше в ДОС.
Система безопасности (списки пользователей, права доступа)	Является частью приложения	Список пользователей и права доступа являются отдельными документами. Может быть несколько списков пользователей.	Реализация проще в ФОС. Удобство использования лучше в ДОС.
Поиск документов	Является частью приложения	Поиск реализуется отдельным документом. Может быть реализовано несколько типов поиска. Настройки поиска сохраняются в документах.	Реализация проще в ФОС. Удобство использования лучше в ДОС.
Печать документов (отчетов)	Является частью приложения. Отчетные формы могут создаваться дополнительно.	Реализуется отдельным документом. Отчет может быть получен как результат взаимодействия нескольких документов.	Реализация проще в ФОС. Удобство использования лучше в ДОС.

проводится до завершения работы над всей системой. Количество интерфейсов конечно и способы их взаимодействия могут быть реализованы заранее.

В том случае, если система расширяется с течением времени, добавляются новые типы документов, то необходимо заранее проработать следующие вопросы:

- как взаимодействуют независимые документо-ориентированные интерфейсы, которые «не знают» друг друга, ведь возможностей базового интерфейса чаще всего ограничены;
- как осуществляется взаимодействие «один ко многим» документо-ориентированных интерфейсов;
- как избежать заикливания цепочки событий.

Предлагается использовать следующие способы взаимодействия:

1. Наличие универсальной системы сообщений на основе стандарта CMIS [3]. Реализуется система подписки на события. Также данная система помогает избежать заикливания, так как отслеживается вся цепочка событий.
2. Взаимодействие системы и документо-ориентированных интерфейсов реализуется на скриптовом языке с возможностью модификации исходных кодов. Тем самым значительно упрощается создание основного приложения информационно-управляющей системы.

Выводы

На основе проведенного анализа можно сделать вывод, что функционально-ориентированную систему значительно проще реализовать. Удобство использования же лучше в документо-ориентированной системе, хотя и значительно повышается сложность реализации. Следует обратить внимание еще на ряд факторов.

Сложность реализации и надежность. Документо-ориентированная система в значительной степени распределяет сложность реализации по различным типам документов, из-за чего повышается и надежность.

Привязка к языкам программирования. В документо-ориентированной системе базовый интерфейс является абстрактным. Для написания модулей может использоваться любой язык программирования. В реализации автора используется архитектура COM (Component Object Model), использовались Visual C++, Delphi, C#. Различные модули, реализующие типы документов, имеют как двух уровневую, так и трех уровневую архитектуру (с использованием сервера приложений).

Расширение системы. Документо-ориентированная система предоставляет неограниченные возможности по расширению. Создается новый модуль, реализующий новый тип документа, и подключается к системе.

Предложены способы взаимодействия независимых документо-ориентированных интерфейсов. Данные способы позволяют значительно сократить объемы работ по созданию базовой функциональности информационно-управляющей системы.

Практическая реализация архитектуры документо-ориентированных систем ведется автором в собственных системах DocsDale и QDocs. Разработка этих систем ведется с 2004 года. Данные информационно-управляющие системы предназначены для автоматизации деятельности вуза при обеспечении учебного процесса. Созданы

программные модули, реализующие типы документов: учебные планы, расчет нагрузки, распределение нагрузки, штатное расписание, студенческие группы, шаблоны документов, общие справочники, редактор отчетов, общие файлы и другие. Работа над системами продолжается.

Список источников

- [1] SDKs and references for Office developers. Электронный ресурс. Режим доступа: <http://msdn.microsoft.com/en-us/office/aa905496.aspx>
- [2] Cocoa Document Architecture (CMIS). Электронный ресурс. Режим доступа: http://developer.apple.com/library/mac/#documentation/Cocoa/Reference/ApplicationKit/Classes/NSDocument_Class/Reference/Reference.html.
- [3] Content Management Interoperability Services (CMIS). Электронный ресурс. Режим доступа: <http://xml.coverpages.org/cmis.html>.