Технология быстрой подготовки проектов к функциональному запуску

Введение

В настоящее время, все более и более актуальной задачей становится создание программного продукта, позволяющего автоматизировать труд разработчика прикладного программного обеспечения (ППО). Под *разработчиком* в данной статье подразумевается *участник создания ППО на любом его этапе.* По нашему мнению специалист, который является носителем концептуальной модели предметной области, также может и должен участвовать в разработке ППО [соответствующая ссылка на статью] и, следовательно, в некотором смысле является разработчиком.

Обострение потребности в автоматизации труда разработчика ППО в последнее время происходит по следующим причинам:

- Возросла конкуренция на рынке разработчиков ППО для автоматизации деятельности предприятий. От производителей ППО требуется существенно сокращать сроки разработки ППО.
- Бурное развитие информационных технологий. Разработанные программы быстро устаревают морально и технически и их нужно заменять на более совершенные.
- Возрос уровень « информационной культуры» конечных пользователей ППО. Многие предприятия и специалисты получили большой опыт в результате внедрения и эксплуатации ППО.

Современные CASE-средства позволяют автоматизировать практически любой этап создания ППО, но, к сожалению, невозможно говорить о том, что имеется единый программный продукт, способный автоматизировать все этапы разработки. В настоящее время коллективами разработчиков ППО используются следующие инструментальные программные средства: Oracle Designer 2000, S-Designer, Clarion, Rational Rose. Данные программные продукты предлагают открытую CASE – технологию создания ППО, но, к сожалению, автоматизируют работу только проектировщиков и программистов - кодировщиков. На этапе построения концептуальной модели, с участием в этом процессе пользователей, данные программные продукты (ПП) сложны и недостаточны. Как дополнение к рассмотренным CASE – средствам, можно взять программный продукт Erwin, позволяющий спроектировать и сформировать структуру БД создаваемого ППО. Данный продукт автоматизирует конечный этап деятельности проектировщика системы, начальный этап деятельности прикладного программиста и, впоследствии – справочный инструментарий для администратора БД, но не позволят организовать эффективное участие в разработке носителя концептуальной модели – специалиста, для которого предназначена разработка.

В данной статье предлагается к рассмотрению новое CASE – средство, автоматизирующее и обеспечивающее связь всех этапов разработки ППО начиная от построения концептуальной и информационной модели и заканчивая генерацией программного кода, как клиентской, так и серверной части разрабатываемой системы.

Особенностью рассматриваемого CASE – средства является то, что оно ориентировано на активное участие в разработке носителя концептуальной модели, т.е. специалиста предметной области или аналитика - постановщика задачи.

Ограниченность разнообразия генерируемых форм компенсируется использованием открытой технологии. При желании, можно модифицировать получающиеся формы или(и) добавить новые шаблоны, т.о. расширять класс генерируемых объектов.

Назначение и возможности предлагаемой технологии

Целью предлагаемой технологии является снижение затрат, уменьшение сроков на разработку, и автоматизация деятельности всех участников разработки прикладного программного обеспечения.

Технология быстрой подготовки проектов к функциональному запуску основана на автоматической генерации прототипа ППО по модели предметной области. Прототип ППО представляет собой работоспособное клиент-серверное приложение, которое удовлетворяет основным требованиям, предъявляемым к современным информационным системам, основанным на реляционных СУБД.

Предполагается, что специалисты предприятия, аналитики и проектировщики ППО, начиная с этапа построения концептуальной модели, используют предлагаемый программный продукт для накопления и систематизации знаний по разрабатываемой системе. Результатом начального этапа создания ППО является концептуальная модель будущей системы.

Концептуальная модель предметной области (КМПО) представляет собой формальное описание понятий, их структуры, отношений между ними и операций над ними на наиболее абстрактном уровне. Особенностью концептуальной модели является то, что она не ориентирована ни на какую программную платформу. В концептуальной модели не содержится информации о том, в каких таблицах представлены понятия, какие типы данных используются для представления значений атрибутов понятий и т.п.

Информационная модель предметной области (ИМПО) получается путем отображения КМПО на конкретную физическую модель данных. Таким образом, можно сказать, что ИМПО содержит в себе КМПО или КМПО является подмножеством ИМПО.

На самом первом этапе разработки ППО, КМПО разрабатывается аналитиком (постановщиком задачи) без участия программистов, но при активном участии специалистов - будущих пользователей информационной системы.

На следующем этапе осуществляется проектирование БД путем расширения КМПО до ИМПО. Затем можно производить первую генерацию предполагаемой системы.

На этапе разработки системы, в процессе уточнения постановки задачи, информационная модель претерпевает различные изменения, поэтому процесс генерации ППО носит итерационный характер. Можно рассматривать этот процесс как систему с обратной связью (см. Рис.1)

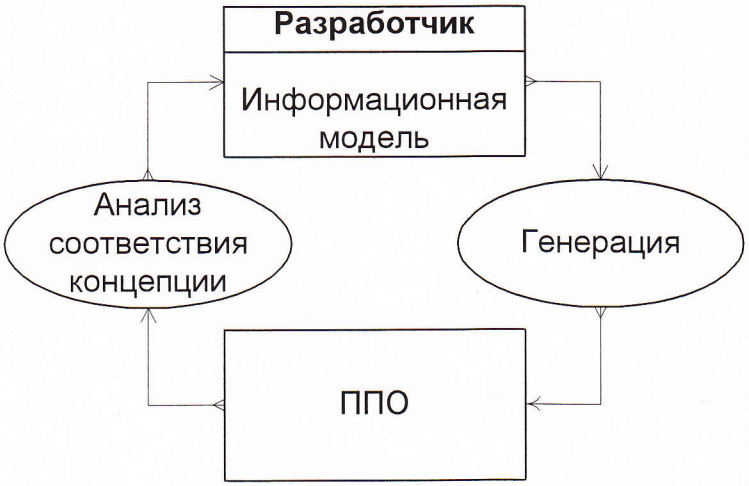

Рис. 1. Итерационный характер создания ППО

Таким образом, в результате последовательных итераций, можно получить ППО, соответствующее сформированному информационному описанию и в точности соответствующее требованиям пользователя.

Рассмотрим пример. На Рис.2 представлена информация об атрибуте «Документ» понятия «Пациент». Все информационные поля разбиты на две группы. Первая группа полей (расположены на верхней панели формы) предназначена для описания данных КМПО. Эти данные инвариантны по отношению к конкретному программному обеспечению, которое будет использоваться для реализации результирующего ППО. Во вторую группу полей (расположены на нижней панели формы) вводятся данные, имеющие отношение к ИМПО.

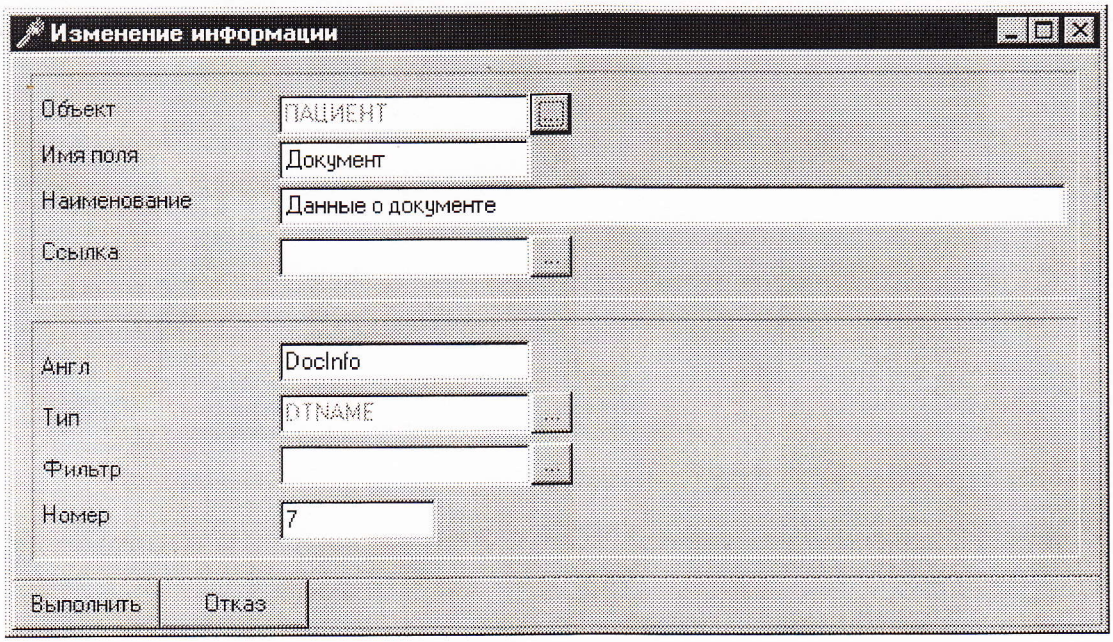

Рис. 2. Данные атрибута ДОКУМЕНТ понятия ПАЦИЕНТ

Необходимо отметить, что в состав КМПО входят разнообразные словари, справочники и классификаторы. Пример экранного интерфейса для формирования справочников представлен на Рис.3.

Справочники

Код	Обозначение	Наименование	Кол-во
024.	ДОМЕНЫ	Домены	10
025.	ПЛЮСМИНУС	Плюс, минус и пусто	3
026.	ТДОКУМЕНТ	Тип документа	10
027.	ТПРОЦЕСС	Тип процесса	2
028.	ТОПЕРАЦИИ	Тип операции	5
029.	ТРЕСУРС	Тип ресурса	7
030.	ТСУБЪЕКТ	Тип субъекта	4
031.	РОЛЬ_ИСП_ОП	Роль исполнителя в операции	3
032.	РОЛЬ_РЕС_ОП	Роль ресурса в операции	2
033.	РОЛЬ_ДОК_ОП	Роль документа в операции	10
034.	ПОЛ	Пол	2
035.	ДОКФИЗЛИЦ	Документ физического лица	2
036.	УЧАСТКИ	Участки	3

Добавить | Изменить | Удалить | Обновить | Элементы

Рис. 3. Словарь данных. Перечень общих справочников

В результате процесса генерации могут быть получены формы, представленные на Рис.4 и Рис.5.

Информация о пациенах

Участок, к которому относится пациент

Фамилия	Имя	Отчество	Дата рождения	Вид документа	Данные о документе	Участок	Адрес проживания пациента
Иванов	Иван	Иванович	03.03.60	ПАСПОРТ	Серия XXI-МЮ N 645302	ЦЕНТР	Москва, Тверская, 17-45
Петров	Петр	Петрович	01.02.72	ПАСПОРТ	Серия XXV-СБ N 204576	СЕВ	г.Москва, ул.Ижорская, 1-80

Добавить | Изменить | Удалить | Обновить

Рис 4. Результат генерации. Экранная форма для работы с данными о пациентах

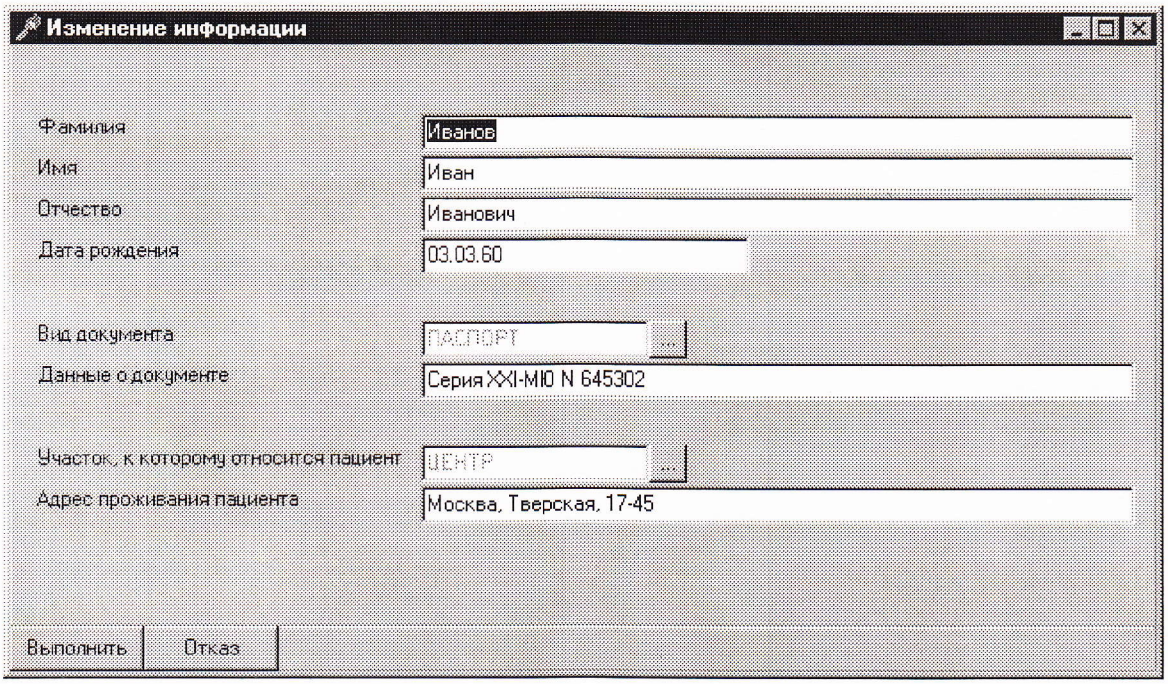

Рис 5. Результат генерации. Экранная форма для ввода и редактирования информации и пациенте

Вполне возможно, что полученный код будет полностью соответствовать требованиям к разрабатываемому ППО, в противном случае, на данном этапе, в обязательном порядке, должен подключаться прикладной программист. Т.к. получаемое ППО представляет собой открытую систему, внесение изменений вполне возможно и даже рекомендуемо. Вероятные изменения - оптимизация доступа к данным и модификация экранных форм. Результатом работы прикладного программиста считаем сформированный проектный файл, который можно легко скомпилировать в исполнимый exe-модуль.

Реализация технологии генерации ППО

Основу технологии генерации ППО составляют пользовательские средства формирования словаря данных, средства генерации ППО на основе интерпретации шаблонов программного кода и структура создаваемого на выходе ППО.

КМПО и ИМПО представляются в словаре данных. По сути своей, вся информация о системе, как то, структура данных, организация данных, экранные формы, отчеты, конфигурация системы хранится в некотором стилизованном словаре данных, обеспечивающем строгую классификацию по типам: таблицы, формы и т.д.

Предлагаемая технология генерации ППО предназначена для решения различных прикладных задач с обязательным участием специалистов в области программирования. Участие специалиста (конечного пользователя) предусматривается на этапе создания концептуальной модели и при анализе получившихся результатов в ходе очередной версии сгенерированного ППО.

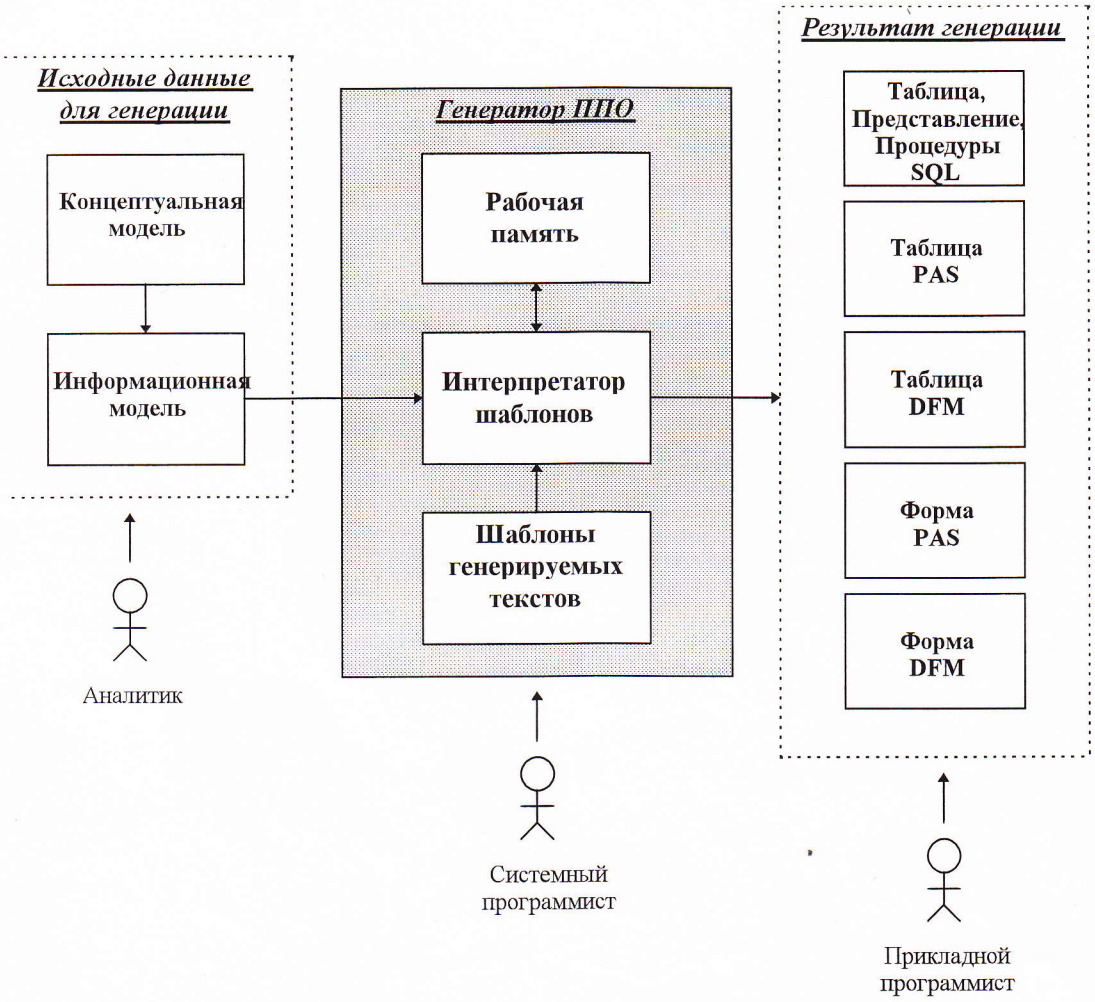

Рис. 6. Схема работы со средствами генерации ППО

На схеме (см. рис.6) проведены границы профессиональной деятельности аналитиков и программистов на очередном итерационном этапе создания ППО. Следует считать эти границы условными но, в идеале, при реализации крупного проекта, действительные границы спецификации будут приближаться к данным. На данной схеме, разделение разработчика на аналитика, системного и прикладного программиста также весьма условно. Определим термином *«системный» программист*, применительно к данному случаю, *разработчика, занимающегося дополнением и модификацией шаблонов генерируемых форм и структуризацией словаря данных, т.е. развитием технологии генерации.* Становится ясно, что участие системного программиста на всех итерационных этапах необязательно. Участие прикладного программиста, который в данном случае выполняет роль кодировщика, также становится необязательным, при достаточной полноте и достаточно развитых средствах генерации ППО. Таким образом можно, в идеале, исключить из данной итерационной схемы системного и прикладного программиста, оставив

лишь аналитика. Под *аналитиком* будем понимать *специалистов, занимающихся постановкой задачи и носителей концептуальной модели - экспертов в заданной предметной области.*

Принципиальная схема функционирования генерируемого ППО показана на Рис.7. Ввод-вывод данных реализован через модуль «Форма». Вывод данных может также осуществляться на принтер или другое графическое устройство вывода. Работа с двумерными массивами данных осуществляется посредством модуля «Таблица». Посредством задания значений в поля, расположенные в «Зоне фильтра» модуля «Таблица» осуществляются стандартные запросы: ограничение выборки и сортировка данных запроса.

Многообразие классов генерируемых объектов не ограничивается только «Таблицей» или «Формой». Открытая система концептуального генератора позволяет расширять и вводить новые классы генерируемых объектов. Таким образом, не ограничиваются функциональные возможности концептуального генератора и открывается большое поле деятельности для автоматизации труда прикладного программиста.

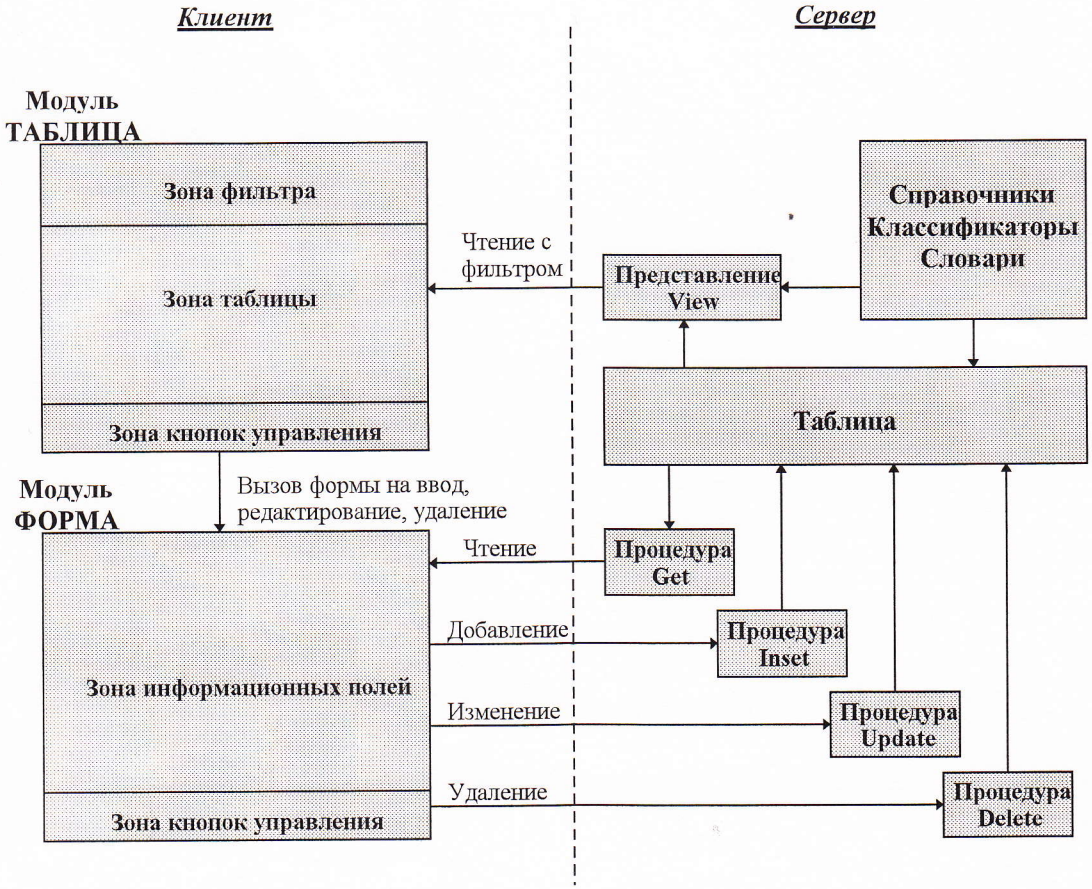

Рис.7. Архитектура генерируемого ППО

Краткая характеристика генерируемого прототипа ППО:

- в качестве серверной части используется СУБД Microsoft SQL Server 6.5
- клиентская часть функционирует под Windows 95/98/NT
- код клиентской части формируется в виде исходных текстов для Delphi 3.0
- прототип ППО позволяет просматривать, выбирать, вводить, редактировать и удалять информацию
- прототип ППО включает средства разграничения прав доступа и аудита действий пользователей

Заключение

В данной статье рассмотрены средства генерации ППО, представляющие собой специальное программное обеспечение, позволяющее путем несложных манипуляций превратить информационную модель будущей системы в готовое предметно - ориентированное ППО, полностью или частично исключающее из процесса создания ППО труд программистов.

Предлагаемая технология позволяет сократить сроки и стоимость разработки ППО, унифицировать программный код разрабатываемого ППО, что существенно упрощает сопровождение и доработку ППО, упорядочить взаимодействие разработчиков ППО на различных этапах разработки.

Актуальность предлагаемой технологии также заключается в ориентации на современные объектно-ориентированные средства программирования и Microsoft SQL Server, который с выходом 7-ой версии все более привлекает к себе внимание предприятий в качестве основы для создания комплексных систем автоматизации.