УДК 004.358:681.518

И.А. БОЖИНСКИЙ

ОБ ОДНОМ ПОДХОДЕ К МОДУЛЬНОМУ ПРОЕКТИРОВАНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ

Предлагается подход к проектированию информационных систем на основе структурно-логического синтеза универсальных модулей. Каждый модуль соответствует одной типовой функциональной компоненте при проектировании информационной системы: от модуля, реализующего тип модели данных, до типовых модулей построения интерфейса пользователя. В качестве интегральной оболочки синтезируемой информационной системы предлагается фреймовая модель.

Введение

Разработка и реализация современных крупных информационных проектов имеет, как правило, затяжной характер, их стоимость превосходит запланированную, а окончательный продукт получается ненадежным и сложным в сопровождении [1]. Это привело к ситуации, которая известна под названием «кризис программного обеспечения». Хотя первые упоминания о кризисе были сделаны еще в конце 80-х годов, даже спустя 30 лет его все еще не удалось преодолеть. В настоящее время многие авторы видят причины этого кризиса в следующем:

– разработка около 40% систем заканчивается неудачно или прекращается до завершения работ;

– рационально интегрировать интересы бизнеса и используемой информационной технологии удается не более чем в 25% проектируемых систем;

– только 20-30% информационных систем отвечают всем критериям достижения успеха.

Основные неудачи при создании программного обеспечения вызваны отсутствием полной спецификации всех требований на этапе проектирования, приемлемой методологии разработки или недостаточной степенью разделения общего глобального проекта на отдельные компоненты, поддающиеся эффективному контролю и управлению. Для разрешения этих проблем в практике проектирования информационных систем был предложен структурный подход, называемый жизненным циклом разработки программного обеспечения.

Под жизненным циклом разработки информационных систем традиционно понимается упорядоченная совокупность этапов, обеспечивающих создание качественного программного продукта. В литературе существует множество нареканий на несовершенство данного структурирования, но смысл его в целом достаточно ясен – борьба со сложностью процесса разработки программных продуктов путем разделения этапов и локализации только тех задач, которые могут и должны решаться именно здесь [2].

Обзор современных методов проектирования информационных систем

Технология программирования уже прошла достаточно долгий путь развития, и сейчас происходит переоценка ее фундаментальных исходных посылок. Первым кандидатом для такой переоценки стала традиционная точка зрения на процесс разработки как на процесс, основанный на понятии жизненного цикла [3].

Существуют различные подходы к разработке информационных моделей и систем. Одно из таких направлений представляет W/O (Warnier/Orr) методология DSSD (Data Structured System Development), которая объединяет методологию Warnier по использованию логических структур данных и логических конструкций программ. Такая методология предполагает, что в распоряжении проектировщика информационной системы имеется представительный набор процедурных шаблонов для широкого класса программируемых задач. В настоящее время DSSD-методология «переросла» из методологии разработки программ в методологию разработки систем.

Следующим подходом к созданию моделей программ является логическое моделирование Гэйна. Логическая модель системы проектируется в процессе последовательного применения таких этапов:

– описание природы предметной области с помощью диаграмм потоков данных (Data Flow Diagrams);

– сведение полученной на предыдущем этапе информации в двумерные таблицы, которые в дальнейшем нормализуются;

– коррекция DFD с учетом результатов нормализации предыдущего этапа;

– разбиение полученной в результате выполнения предыдущих этапов модели на «процедурные единицы», а также определение деталей каждой процедурной единицы.

Третьим в ряду подходов к созданию модели проектируемой программы является метод Иордана. Он включает два компонента: инструментальные средства и методики. Методология Иордана ориентирована на проектирование систем обработки данных. Под инструментальными средствами здесь понимаются различные диаграммы, используемые при описании моделей требований и моделей архитектуры проектируемой информационной системы. Самые известные из таких диаграмм - диаграммы потоков данных DFD. Однако их недостатком является отсутствие средств описания отношений между данными и их «поведения» во времени. Вот почему в инструментальные средства метода Иордана на сегодняшний день кроме DFD включены ERD-диаграммы (Entity Relationship Diagrams) и STD-диаграммы (State Transition Diaggams).

Подход, основанный на методе Иордана, помогает перейти от бланка на бумаге и/или экранной формы к хорошо организованной системной модели. Первоначально эти методики базировались на традиционном top-down проектировании. В настоящее время используется метод событийного разбиения (event partitioning). При этом сначала создается контекстная диаграмма верхнего уровня, где определяются системные ограничения и интерфейсы с «внешним миром». Затем с помощью техники интервью формируется список событий из внешней среды, на которые система должна реагировать. Такой подход обеспечивает простой базис для формирования «сырой» DFD. Несколько DFD-реакций могут быть объединены в редакцию более высокого уровня. Проблемы, возникающие при использовании метода ERD-диаграмм, связаны, прежде всего, с трудностями интеграции компонентов при разработке всей системы. Вот почему в последнее время этот метод был обобщен за счет введения интегрированной базы данных. При этом ERD-метод трансформируется в структурную методологию, основные этапы которой сводятся к разработке ERD [4].

Последним методом, который рассматривается в данном обзоре, является метод структурного проектирования. Структурное проектирование сделалось действительно мощным и активно используемым на практике подходом из-за того, что к моделям и методам были добавлены оценки результатов проектирования. Здесь предлагаются все проектные решения располагать в трехмерном пространстве «содержание – сложность – связность». И утверждается, что хорошими проектными решениями будут лишь те, которые при заданном содержании имеют минимальную сложность и максимальную связность.

Постановка задачи

В связи с изложенными выше тенденциями в проектировании информационных систем предлагается на основании анализа реляционных моделей данных различных прикладных информационных систем разработать типовые структурно-логические модули, соответствующие классам задач проектирования реляционных моделей данных, а также типовые интерфейсы пользователя. Каждый такой модуль включает функционально полную реляционную модель для решения задачи хранения и обработки данных на основе реляционной СУБД (системы управления базами данных) и интерфейс пользователя.

Решение поставленной задачи

В качестве структурно-логической единицы, соответствующей синтезируемой информационной системе, предлагается фреймовая модель. Фрейм – каркас, рамка, который объединяет в себе слоты – необходимые компоненты информационной системы. В общем виде такая модель может быть записана следующим образом:

$$FR\left\{< R_1, C_{11,} C_{12,..} C_{1m,} >,... < R_2, C_{21}, C_{22,} ... C_{2m,} >, < R_{k1}, C_{km} >\right\}, \qquad (1)$$

где *FR* – имя фрейма (идентификатор проектируемой информационной системы); пара *<Ri,Ci>* – *i* -й слот фрейма; *Ri,* – имя слота; *Ci* – значение слота.

Слот в модели (1) является структурно-логической конструкцией для реализации конкретных функций информационной системы. Слоты из фрейма можно удалять, добавлять, изменять их функциональное назначение. Для удобства дальнейших рассуждений слот-компонент фрейма определим как e. Любому компоненту-слоту e будет соответствовать определенная арность $\mathrm{Dim}(e)$, которая показывает число модулей сопряжения. Модули сопряжения компонентов-слотов могут быть описаны набором признаков или параметров.

Таким образом, одним из факторов, определяющих функциональные возможности компонентов-слотов каждого класса, является наличие и типы встроенных модулей сопряжения. Другим фактором является различная программная реализация компонентов-слотов и тип потребляемого в режиме функционирования ресурса. Конкретной программной реализации компонента-слота e функциональной элементной базы соответствует его представление вектором параметров $V(e)$.

Любой компонент-слот e функциональной базы можно представить следующими образом $e=[I(e),G(e),V(e)]$, где $I(e)$ – номер класса компонента-слота; $G(e)$ – тип структуры модулей сопряжения; $V(e)$ – вектор конструктивных параметров.

Для формального определения возможных функциональных вариантов компонентов-слотов введем преобразование подобия s, отображающее множество E на себя $s: E \to E$. Два образующих компонента-слота e_1 и e_2 подобны, если существует некоторое преобразование подобия $s \in S$, такое, что $se_1 = e_2$. Множество S содержит среди прочих преобразование подобия вида s_0, для которого $s_0 e = e$. Конкретный вид преобразований подобия зависит от проектируемого семейства информационных систем.

Будем считать, что используемые преобразования подобия удовлетворяют следующим требованиям:

– множество S преобразований подобия s является группой;

– любое $s \in S$ отображает класс E_i в себя при любом номере класса компонентов-слотов;

– элементы множества S не влияют на структуру модулей сопряжения, но могут влиять на их признаки.

Таким образом, структура модулей сопряжения и индекс класса компонентов инвариантны относительно преобразования подобия.

С учетом свойства инвариантности для компонента-слота можно ввести понятие функционального класса компонента-слота: $z=[I(e),G(e)]$, которое отражает наиболее важную информацию о компонентах-слотах, заключенную в номере класса и структуре модулей сопряжения. Для функциональной базы E можно определить интегральную категорию верхнего уровня – функциональную базу типов. Рассмотрим отображение $h: E \to E'$, где E и E' – две функциональные базы одного и того же семейства, связанные с одной и той же группой преобразований подобия S.

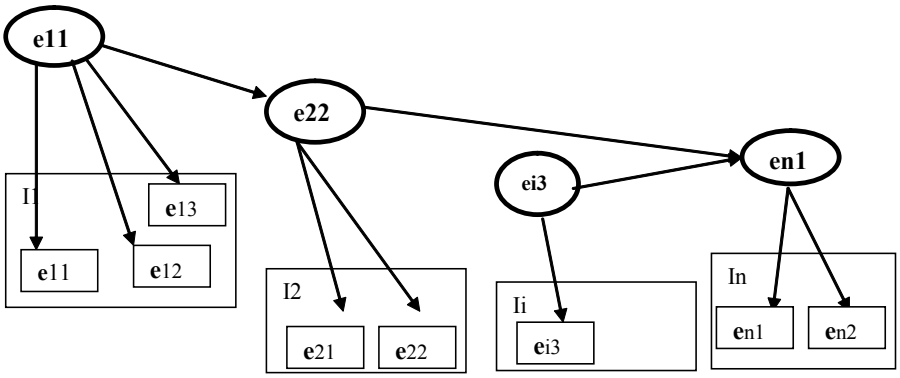
Будем называть преобразование h инвариантом связей, если для любого компонента-слота $e \in E$ компоненты e и $e' = h(e)$ имеют одни и те же программные модули сопряжения, а также $se \to se'$ для $\forall s \in S, \forall e \in E$. С помощью отображения h можно осуществить переход от функциональной базы E к новой элементной базе E', не нарушая общей структуры информационных систем, в которых используются компоненты E.

Данный случай возникает при замене имеющейся функциональной базы семейства элементной базой нового поколения. Структура элементной базы семейства формируется классами компонентов-слотов. Внутри класса слоты отличаются структурой модулей

сопряжения $G(e)$. Компоненты одного класса, отличающиеся какими-либо из перечисленных выше характеристик и признаков, будем называть модификациями компонентов-слотов класса. Модификации компонентов учитываем различными номерами компонентов; так, при двухсимвольном обозначении e_{ij} первый символ указывает класс компонента-слота, второй – номер модификации.

Компоненты класса, связанные с другими его компонентами одним и тем же преобразованием подобия S, являются универсальными. Число универсальных компонентов-слотов в классе определяется числом используемых преобразований подобия.

На рисунке представлена структурная схема унифицированной модульной системы синтеза информационной системы на основе структурно-логических элементов. **I1** – блок унифицированных типовых реляционных схем баз данных; **I2** –блок унифицированных типовых интерфейсов пользователя, который состоит из модулей: **e**21 – «главная кнопочная форма», **e**22 – «ленточная форма» и т.д. **In** – блок унифицированных типовых выходных форм пользователя.



Структура универсальных модулей для синтеза информационных систем

Выводы

Предложен подход к синтезу типовых информационных систем основанный на унифицированных структурно-логических модулях, соответствующих классам задач проектирования реляционных моделей данных, а также типовых интерфейсах пользователя. Каждый такой модуль включает функционально полную реляционную модель, для решения задачи хранения и обработки данных на основе реляционной системы управления базами данных и интерфейс пользователя. Предлагаемый подход позволит значительно увеличить эффективность процесса проектирования информационных систем на основе реляционных баз данных.

Дальнейшие исследования предполагается проводить в направлении использования предложенных структурно-логических модулей для построения автоматизированных алгоритмов синтеза информационных систем.

Список литературы: 1. *Пасічник В. В.* Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. К. : BHV, 2006. 386 с. **2.** *Филатов В. А.* Методы и средства проектирования информационных систем и распределенных баз данных / В. А. Филатов, Р. В. Семенец // Вестник Херсонского национального технического университета. 2007. № 4(27).С. 203-207. **3.** *Филатов В. А.* Концепция проектирования средств информационной поддержки в системе государственного управления / В. А. Филатов, Е. И. Фалькович // Збірник наукових праць Української Академії державного управління при Президентові України. Харків. 2001. Вип. 1. С. 68-70. **4.** *Йордан Э.* Структурное программирование и конструирование программ: Пер. с англ. М.: Мир, 1979. 415 с.

Поступила в редколлегию 11.10.2012

Божинский Иван Андреевич, канд. техн. наук, зам. начальника НИЧ ХНУРЭ. Научные интересы: информационные системы. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-378.