

МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

Розглянуто особливості проектування інформаційних систем та розподілених баз даних із застосуванням сучасних технологій проектування. На основі аналітичного огляду існуючих підходів до проектування інформаційних систем зроблено висновок про необхідність застосування ефективних моделей проектування в сучасних CASE-засобах. Розроблено концептуальну математичну модель процесу проектування розподілених корпоративних інформаційних підсистем. Наведено перелік етапів проектування у термінах {[Ціль етапу], [Опис етапу], [Метод рішення], [Результат]}.

Введение.

В настоящее время ключевая роль в достижении успеха большинства компьютеризированных систем принадлежит программному обеспечению. В последнее десятилетие прикладные программы проделали путь от маленьких и сравнительно простых приложений из нескольких строк кода до очень больших и сложных приложений, состоящих из нескольких миллионов строк. Многие из этих приложений требовали постоянного сопровождения, включая исправление выявленных ошибок, реализацию новых требований пользователей, а также перенос программного обеспечения на новые или модернизированные вычислительные платформы. Усилия и ресурсы, затрачиваемые в настоящее время на сопровождение программного обеспечения, возрастают очень высокими темпами.

Целью работы является исследование и разработка эффективных моделей и методов проектирования информационных систем и распределенных баз данных.

1. Основные проблемы проектирования информационных систем

Разработка и реализация многих современных крупных информационных проектов имеет, как правило, затяжной характер, их стоимость превосходит запланированную, а окончательный продукт получается ненадежным и сложным в сопровождении. Все 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 [1]. В настоящее время многие авторы называют этот кризис «программного обеспечения», некоторые причины которой состоят 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% систем;
- только 20-30% информационных систем отвечают всем критериям достижения успеха.

Основные неудачи при создании программного обеспечения были 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 разбиения, но смысл его в целом достаточно ясен – борьба со сложностью процесса разработки программных продуктов за счет структуризации этапов и локализации на каждом из них только тех задач, которые могут и должны решаться именно здесь. Если перечисленные выше этапы укрупнить, останутся стадии: проектирования, реализации и сопровождения.

Проектирование предполагает составление формальных и/или формализованных спецификаций. Реализация – преобразование этих спецификаций в программный код (автоматическое и/или автоматизированное). Сопровождение – тестирование разработанной системы, ее внедрение и последующую модификацию, которая, в свою очередь, может вернуть весь процесс к стадии проектирования или перепроектирования.

2. Методы проектирования информационных систем и распределенных баз данных

Существуют различные подходы к разработке информационных моделей и систем. Одно из таких направлений представляет W/O (Warnier/Ort) 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. При этом явно выделены уровень программирования (programming level) и системный уровень (system level). Основу подхода составляют два концептуальных представления – диаграммы входов (Entity diagrams), обеспечивающие определение системного контекста, и модифицированные W/O-диаграммы (Assembly-line diagrams), специфицирующие функциональное развитие системы.

Следующим подходом к созданию моделей программ (по идее достаточно близким к W/O-методологии) является логическое моделирование Гэйна. Сам метод ориентирован на создание систем обработки данных. Логическая модель системы проектируется в процессе последовательного применения следующих этапов:

- описание природы предметной области с помощью диаграмм потоков данных (Data Flow Diagrams);
- выделение первичной модели данных (списка элементов данных в каждом информационном узле);
- проверка того, что DFD действительно отражает структуру данных, хранимых в системе;
- 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.

После выполнения этих этапов принимается решение о необходимости прототипирования системы на целевом языке.

И, наконец, третьим в ряду подходов к созданию модели проектируемой программы является метод Иордана [2]. 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 Diagrams).

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 реакцию более высокого уровня. Критерием такого объединения является наличие узлов, связанных общими данными. По существу, событийное разбиение есть не что иное, как метод объектно-ориентированного проектирования.

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.

Последним методом, который рассматривается в данном обзоре, является метод структурного проектирования. Структурное проектирование сделалось действительно мощным и активно используемым на практике подходом за счет того, что к моделям и методам были добавлены оценки результатов проектирования. Здесь предлагаются все проектные решения располагать в 3-х мерном пространстве «содержание – сложность – связность». И утверждается, что хорошими проектными решениями будут лишь те, которые при заданном содержании имеют минимальную сложность и максимальную связность. По существу, этот критерий отражает уже обсуждавшиеся выше понятия абстрагируемости, модульности, связности и т. п. Конкретным примером воплощения структурного подхода является «водопадная» или «каскадная» модель разработки информационных систем.

В качестве примера смены парадигм проектирования можно отметить спиральную модель процесса разработки, предложенную Бозом, суть которой состоит в том, что отдельные фазы процесса «прокручиваются» на постепенно повышающихся уровнях иерархии. Результатом разработки при этом становится не только программный код, но и его представление на более высоком уровне, что особенно важно в тех случаях, когда информационная система и программное обеспечение эволюционируют.

Естественно, что такая модель предполагает наличие более гибких механизмов взаимодействия между потребителями и разработчиками соответствующих инструментальных средств. В Европе это направление ассоциируется, прежде всего, с проектированием на основе объектно-ориентированных методологий, среди которых наиболее популярными являются технология MERISE и методология Шлайера-Мэллора (Shlaer-Mellor).

В конце 90-х годов XX века был разработан Унифицированный язык моделирования (Unified Modeling Language, UML), который является объектно-ориентированным графическим языком для визуализации, спецификации, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению [3].

Несмотря на свои достоинства, UML – всего лишь язык и является одной из составляющих процесса разработки программного обеспечения. Хотя UML не зависит от моделируемой реальности, лучше всего применять его, когда процесс моделирования основан на рассмотрении прецедентов использования и является итеративным и пошаговым, а сама система имеет четко выраженную архитектуру. Неоспоримой заслугой UML является введение понятия прецедента использования. «Экземпляр юзкейса» (сценарий) – это последовательность действий, выполняемых в системе, которые позволяют получить результат, наблюдаемый для актера этой системы, а юзкейс определяется как совокупность всех своих сценариев. Однако UML, вводя это понятие, не дает четких способов применения юзкейсов. Получается, что юзкейсы интуитивно понятны аналитикам, но их применение в похожих ситуациях зачастую приводит к различным результатам [4].

3. Модель процесса проектирования системы управления информационными ресурсами распределенной вычислительной системы

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. Единодушие специалистов состоит в том, что данная точка зрения должна быть заменена такой, которая в большей степени соответствовала бы разработкам, поддерживаемым автоматизированными средами [5].

В связи с изложенными выше тенденциями в проектировании информационных систем, математическую модель процесса проектирования можно представить в виде итерационного процесса, состоящего из отдельных взаимодействующих проектных процедур трех множеств:

- множества моделей $M = \{M_1, M_2, \dots, M_n\}$;
- множества операций над моделями $O = \{O_1, O_2, \dots, O_n\}$;
- множество критериев проектирования $K = \{K_1, K_2, \dots, K_n\}$, где n – количество этапов проектирования.

Весь процесс проектирования представляется последовательностью проектных процедур $\rho = \{\rho_1, \rho_2, \dots, \rho_n\}$. Каждая проектная процедура $\rho_j = \langle O_j, M_j, K_j \rangle$ переводит модель проектируемого объекта в следующее состояние:

$$\rho_1 : M_1 \rightarrow M_2, \rho_2 : M_2 \rightarrow M_3, \dots, \rho_n : M_n \rightarrow M_{n+1}. \quad (1)$$

При этом должны выполняться условия:

$$\forall M_j \in M \exists \rho_j : M_j \rightarrow M_{j+1}, \quad (2)$$

где $M_j \Rightarrow M_{j+1}$, $K_j \subseteq K_{j+1}$, $O_j \subseteq O$, $j = \overline{1, N}$, где \Rightarrow символ предшествования.

Выполнение определенного числа проектных процедур ρ_{ij} на i -ом витке проектирования спиральной модели процесса позволяет перейти на следующий, более высокий уровень данного проектного решения [6]. В соответствии с (1)-(2) рассмотрим модель проектирования информационных систем и распределенных баз данных. Представим основные этапы проектирования информационных систем в терминах:

ЭТАП = {[ЦЕЛЬ ЭТАПА], [ОПИСАНИЕ ЭТАПА], [МЕТОД РЕШЕНИЯ], [РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ЭТАПА]}.

3.1. Формирование проекта.

Цель этапа.

Выбор информационной технологии решения поставленной задачи.

Описание этапа.

На этапе формирования проекта определяется выбор и обоснование методов и средств решения поставленной задачи.

Метод решения.

Системный анализ. Мозговой штурм, круглый стол специалистов в области информационных технологий.

Результат выполнения этапа.

Результатом выполнения этапа должно быть мотивированное решение в пользу выбора той или иной информационной технологии для решения поставленной задачи.

3.2. Анализ предметной области

Цель этапа.

Разработка и анализ информационной модели предметной области.

Описание этапа.

От качества системного анализа на этапе проектирования непосредственно зависит уровень разрабатываемой системы, ее соответствие требованиям пользователя. Последовательность выработки требований пользователя по данному этапу:

- обследование и системный анализ предметной области;
- обобщение результатов системного анализа и создание предварительной концепции информационной системы или распределенной базы данных;

На этапе системного анализа и проектирования может быть рекомендован следующий порядок работ:

- получить модель бизнес и информационных процессов;
- выявить узкие места в бизнес-процессах и наметить пути их ликвидации;
- создать информационную и функциональную модель;
- сформировать список требований к новой информационной системе;
- выбрать методы и средства проектирования и реализации информационной системы;
- сформировать архитектуру системы;
- составить предварительный план проектирования и реализации базовой версии информационной системы;
- оценить трудозатраты разработки новой информационной системы;
- составить технико-экономическое обоснование.

В процессе выполнения работ, ввиду особой важности этапа, необходимо обратить внимание на следующие моменты:

- большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними, требующие моделирования и анализа данных и процессов;
- наличие тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования, например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, приложений аналитической обработки (поддержки принятия решений), использующих нерегламентированные запросы к базам данных большого объема;
- необходимость интеграции существующих и вновь разрабатываемых приложений; функционирование в неоднородной среде на нескольких аппаратных платформах.

Метод решения.

Функциональное моделирование.

Результат выполнения этапа.

Концептуальная модель системы, состоящая из описания предметной области, ресурсов и потоков данных, перечень требований и ограничений к технической реализации.

3.3. Спецификация задачи.

Цель этапа.

Разработка и формализация информационно-логической модели взаимосвязанных задач. Форма представления: граф-схемы, сети Петри.

Описание этапа.

На основании концептуальной модели разрабатывается информационно-логическая схема взаимосвязанных задач пользователя.

Метод решения.

Функциональное моделирование.

Результат выполнения этапа.

Информационное обеспечение системы: структуры задач, условия выполнения, межзадачные связи.

3.4. Создание приложения.

Цель этапа.

Разработка и реализация приложения инструментальными CASE-средствами проектирования информационной системы.

Описание этапа.

При помощи интерактивного интерфейса CASE-средств автоматизированного проектирования осуществить генерацию и запуск приложения.

Метод решения.

Работа в среде программных средств.

Результат выполнения этапа.

Функционирующее приложение.

Выводы.

На основе анализа особенностей проектирования информационных систем и распределенных баз данных рассмотрены основные проблемы, стоящие перед специалистами в области проектирования на современном этапе. Предложена модель процесса проектирования распределенных информационных систем. Рассмотрены этапы проектирования в терминах ЭТАП={[ЦЕЛЬ ЭТАПА], [ОПИСАНИЕ ЭТАПА], [МЕТОД РЕШЕНИЯ], [РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ЭТАПА]}.

ЛИТЕРАТУРА:

1. Фалькович Е.И., Филатов В.А. Концепция проектирования средств информационной поддержки в системе государственного управления // Збірник наукових праць Української Академії державного управління при Президентіві України. Харків. – 2001. Вип. 1. С. 68-70.
2. Йордан Э. Структурное программирование и конструирование программ: Пер. с англ. М.: Мир, 1979. 415 с.
3. Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений: Пер. с англ. М.: ДМК Пресс, 2002. 698 с.
4. Фридман А. Л. Основы объектно-ориентированной разработки программных систем. - М.: Финансы и статистика, 2000.
5. Пономаренко Л. А. Програмні агентні технології в адмініструванні баз даних / Л. А. Пономаренко, В. О. Філатов. // Вісник Київського торговельно-економічного університету. – 2001. – №3. – С. 68–73.
6. Филатов В.А. Модель поведения автономного агента на основе теории автоматов / В.А. Филатов // Вестник Херсонского государственного технического университета. – Херсон: ХГТУ. – 2004. – №1(19). – С. 108 – 111.

Филатов Валентин Александрович

Доктор технических наук, профессор кафедры Искусственного интеллекта Харьковского национального университета радиоэлектроники Filatov_val@ukr.net

Научные интересы: базы данных и знаний, агентные технологии, мультиагентные системы, извлечение знаний из данных.

Семенец Руслан Васильевич

Инженер Научно-исследовательской лаборатории системных технологий Харьковского национального университета радиоэлектроники Ars_sem@ukr.net