

Александр Сердечный

**Как избежать
"столпотворения"
в ИТ (IT) и
программировании**



Питер Брейгель Старший
Вавилонская башня (фрагм.), Роттердам

Александр Петрович Сердечный

Как избежать «столпотворения» в ИТ (ИТ) и программировании

*http://www.litres.ru/pages/biblio_book/?art=73494643
SelfPub; 2026*

Аннотация

В брошюре, с целью преодоления проблемы разрозненного хранения и представления данных в БД, проблем модификации ПО, а также усиливающейся разобщённости языков программирования – предлагаются и описываются концепции:

- единой для всех унифицированной гибкой структуры баз данных;
- параллельной потоковой обработки данных, позволяющей выполнять постепенную модификацию алгоритмов в режиме онлайн;
- единого универсального функционального языка программирования от уровня почти "машинных" команд и до сколь угодно абстрактного уровня объектно-ориентированных языков.

Содержание

Угрожающие успехи ИТ	4
Ноу-хау: унифицированная база данных	8
Конец ознакомительного фрагмента.	11

Александр Сердечный

Как избежать "столпотворения" в ИТ (IT) и программировании

Угрожающие успехи ИТ

NB. При цитировании ссылка на автора и статью обязательна. Связаться с автором можно по почте: cardiac@yandex.ru

Сфера информационных технологий давно уже разделилась по отраслям и в каждой нарабатаны свои технологии, системы программирования и классы объектов. Взять хотя бы бухчѐт с его почти безраздельной монополизацией языком программирования "1С". Есть и отрасли курирующие системы управления производством, и системы бизнес-менеджмента, и т.д.

И везде свои особые средства программирования, классы объектов и даже платформы исполнения. Всѐ это не легко в изучении и создаѐт объективные сложности освоения профессии, обеспечивая ей престижный статус, соответствующий

щую зарплату и востребованность, что весьма устраивает ИТ специалистов. И не секрет, что сейчас именно сфера обучения ИТ является самой востребованной и процветающей.

Процессы разделения и обособления естественны и поддерживаются из престижных и меркантильных соображений. А тем самым цивилизационное *информационное пространство становится всё более фрагментарным и разобщённым.*

Но вспомним, что случилось с проектом строительства "вавилонской башни"? Он рухнул, когда строители стали разговаривать на разных языках. А ситуация с ИТ уже похожа на "столпотворение" – синоним: "бестолковая суета".

Для упорядочивания номенклатуры пытаются создавать десятичные классификаторы – но не всё кратно 10, а на примере библиотечного УДК видим, что основная часть описания его правил посвящена исключениям из них.

Данные по критериям общности группируются в таблицы. Например, паспортные данные у всех будут в таблицах почти одинакового формата, но многие другие будут весьма отличаться на разных предприятиях, даже по составу и количеству таблиц.

Сами же совокупности таблиц предприятий и отраслей группируются по базам данных, причём зачастую различных форматов курируемых разными системами управления (СУБД).

Сейчас почти безраздельно принята концепция процесс-

ной обработки данных, при которой запускается ведущая программа, которая по ходу алгоритма инициирует запуск сервисных процессов получения требуемых данных. Включены также элементы ситуативного программирования, когда нажимаются клавиши мышки и т.п. А управление процессами, очередями и ресурсами выполняет операционная система (ОС).

И ведущая программа, и сервис, и базы данных вынуждены перманентно модифицироваться по мере изменения и развития алгоритмов управления бизнес-процессами и пр. С ростом сложности взаимосвязей это становится почти неразрешимой задачей.

Тем более, что в объектно-ориентированных языках программирования свойства классов объектов и методы работы с ними заложены (из библиотек) фактически в текстах программ, отследить использование которых весьма сложно. Хотя по сути, этим объектам и методам более уместно было бы наличествовать в базах данных.

Итак, проблемы интеграции:

- несовместимость форматов представления данных в базах данных;
- сложность перманентно необходимой модификации процессов и программ;
- разнообразие языков программирования.

Ведомства, монополизировавшие информационные ресурсы, не заинтересованы, да и объективно не способны опера-

тивно осуществлять прогресс в ИТ.

И чтобы решить проблемы интеграции и обмена данными, нужна определённая и согласованная работа. Более подробно эти темы описаны в части V книги [1].

Критикам, как правило, пеняют: "Если критикуешь – предлагай!" Далее и предлагаю.

Ноу-хау: унифицированная база данных

Опытный вариант унифицированной базы данных (УБД) был опробован при разработке программы делопроизводства для городской (3 р-на) мэрии. Программировалась она в Delphi, а база данных в Access. И несколько лет эксплуатировалась без всяких нареканий.

Суть УБД в том, что её структура неизменна и не зависит от типа и организации связей хранящихся в ней данных. Каждое данное хранится отдельно и одинаковым образом. *А тип данного, его назначение и онтологическая суть, подробно описаны в метаданных.*

Метаданные широко используются и содержат всевозможные сведения и о базах данных, и о свойствах и назначении их содержимого, журнал проделанных с базами работ и пр. Но в оперативной работе с данными они обычно не участвуют.

Здесь же предполагаются метаданные, которые характеризуют назначение и форматы данных, а также указывают средства их отображения и способ получения из других данных базы.

В УБД разные данные могут объединяться в одном объ-

екте, составляя набор столбцов виртуальной таблицы. А их значения, относящиеся к одному и тому же экземпляру объекта, составляют строку виртуальной таблицы.

Метаданные для каждого типа данного (вида документа), являющиеся его *онтологическими характеристиками*, должны быть представлены в ней же и содержать текстовое описание, отношения с другими данными и алгоритм получения нового документа этого типа из данных других документов.



Таких таблиц, по-разному группирующих данные, может быть одновременно любое количество. Данные, следовательно, организованы в сетевую структуру. А далее с этими виртуальными таблицами можно выполнять такие же запросы, что и с обычными.

Поскольку данные различного типа (строки, длинные тексты, дата-время, целые и реальные числа разной разрядности) требуют, например в реляционных БД, различного вида полей, то для каждого их типа может быть организован свой массив данных. Их совокупность отмечена бежевым цветом.

Для работы с данными, например с файлами мультимедиа, понадобятся конкретные сервисы. *Должна быть описана форма представления типа данного в пользовательском интерфейсе и указан нужный инструментарий.* Соответствующий сервис тоже может быть автоматически загружен из УБД.

Структура УБД показана ниже. Названия ключевых полей таблиц выделены цветным жирным шрифтом. Любой объект может быть в любой момент назначен компонентом более интегральной структуры. Зелёные соединительные линии обозначают связи при формировании новых документов, красные – при выборке из них данных.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.