成熟的软件组件都是老板用大把、大把的钱堆出来烧出来的, 以最简单的数据库访问组件为例

作者：qmmcu

原文链接：https://ld246.com/article/1519715118189

来源网站：链滴

成熟的软件组件都是老板用大把、大把的钱堆出来烧出来的，以最简单的数据库访问组件为例 自己虽不属于技术强的那类人物，但算是勤奋用功“刨根问底”类型的，总喜欢把一个东西研究个透彻的那，否则心里不踏实，总是云里雾里，废话不多说。

[B/S]

1：在宁波宇泰软件公司时，需要做一个ASP.NET的EIP项目，从PHP版本的PostNuke改版的任务，在VB.NET版本的DotNetNuke也是很出名的，由于不喜欢VB.NET，就彻底放弃研究这个了，由于很的时候接触了国外网上的知名开源软件项目，所以对今天的工作也很有帮助，思想一直没被国内的软项目影响，也一直坚持国外的项目的严谨思路、严谨的架构、高质量的代码编写习惯。那时花费了2月多，就弄出来了，虽然只能连接 SQLServer，对什么注入攻击啥的，也不是很了解，但是用用还是好用，陆续也有2-3个同事用这个框架开发程序，后来有一个开发小组都用这个架构的方式开发EIP，然他们人也会有不断改进的。

[B/S] 2：到杭州浙大快威工作，就是现在的浙大快威电力事业部，由于公司产品用Oracle比较多，所也跟着把数据库访问组件再一次升级、更新，那时候比较喜欢用OleDB方式连接数据库，这样写的代差别不大，但是这个可以同时运行在多种数据库上，兼容性好，数据库都有相应的OleDB的驱动就可了，也比较省事，那时候也不懂设计模式什么的，到底这些数据库驱动方式有啥差别? ODBC? BDE? LEDB? ADO.NET? Oracle.NET 到底有啥差别，也是云里雾里。

[B/S] 3：后来到上海索恩软件宁波分公司工作做日本外包项目，在上海见识到日本NEC公司的项目，尖高手架构.NET，对数据库的事务、并发的严谨高要求，对UML的深入理解，对软件质量的要求、软件项目进度的控制、分工合作等，又有了更高的认识，看日本鬼子的软件那么严谨，那么精密，事又把自己的数据库访问组件彻底进行了优化，支持严格的数据库事务控制、支持严格的并发控制等，是感觉写得越来越庞大、程序越来越多、问题越来越复杂，修正一个错误往往牵涉的面也很广了，经痛苦的折磨，总算又改进为支持事务、并发的处理要求了，感觉自己也提高了一个层次。

[B/S + C/S]

4：自己创业后发现，自己很难接到大型软件项目，大部分是小项目甚至是微型项目、小网站，网络拟机上托管的网站类项目比较多，这时把程序又进行了一次，否则理论与实际是脱钩了，总需要解决存问题，蛤蟆也得吃，只能硬着头皮改进为支持桌面数据库Microsoft Office Access，这样不用装个大的数据库了，而且支持数据库的虚拟机费用也贵一些，折腾起来也不方便，而且大部分不支持SQL Srver, 也不支持ASP.NET, 自己买个服务器放到网上当时也没那个实力，现在放上10个8个也不是大问了，当时钱财也不是很多，有些郁闷阶段，当然给其他公司做外包，也不能想用自己的数据库连接组就可以用自己的组件，有时候也派不上大用处，感觉瞎折腾了很多东西一样，客户也根本不在乎事务也不在乎并发问题等。

[C/S]

5：到了宁波东蓝科技、大部分项目又是用Oracle的，而且对设计模式、并发等的控制要求比较高一，这时也参考了微软的SQLHelp等开源程序，公司里也有深入了解设计模式的高手指点，又把程序改为符合设计模式理念的程序、程序的质量又提高了一个层次，又一次飞跃，同时由于开发人员也多一，开发环境也较好，可以安心写程序，所以把并发方面出现的问题都进行了修复，也对一些算法进行优化，也算是进行了有规模的软件项目开发工作，也做了一个远程数据库对象，提供在客户端直接控数据库的方法，虽然没有得到实际应用，但是证明我当时的思路还是对的，微软现在都有这个内部组了。

[B/S + C/S]

6: 到了杭州东蓝科技，里面有一个写程序拿来主义高手，写程序的思路严谨工作效率也高、他用 Disc z nt，这个是用最新的 ADO.NET 2.0 架构的，虽然在数据库事务、并发控制方面很弱，但是非常适做那些没有严格数据库事务要求的WEB项目，而且里面的命名也很规范，我想写这个代码的人的水平是绝对顶呱呱的，反复研究学习了各个版本后，又用了一个月时间，又把数据库访问组件改进了一下改进为符合ADO.NET2.0的优化方式，代码更简洁了很多，看起来也赏心悦目了，心理也爽了，啥时微软又出个ADO的升级版本，更搞死人了，公司也有一个项目是用了MYSql的数据库，所以把数据库问组件，在MYSql上又做了一个彻底的测试改进，也改进优化了很多环节，同时也支持动软代码生成的兼容性，同时改进优化为支持分布式的数据库架构、同时连接多个多种类的数据库，使这个组件变

更加强大。

[B/S + C/S]

7: 年后做了B2C的网上商城项目后，又有些变化，以前是以开发内部管理系统为主，并发访问的人数不是很多，现在做了B2C的项目后，每天访问的量会有几千人到几万人不等，对数据库并发性能的要更高，又发现了一些错误，在超多用户访问时还是遇到了并发问题、只能马上就修正好，否则怎么对户有交代啊？这脸面就会全丢了，做内部管理系统与做外部网站还是不一样的，还是需要很多经验积，需要不断摸索才能稳定成熟。

[B/S + C/S]

8：现在在浙大网新易盛打工，懒得让别人用这个数据库访问组件了、何必给自己找麻烦呢、谁愿意麻烦就惹吧，要学会低调做人，麻烦都是自己惹的，认可我的就用吧，也很乐意提供技术支持，不认信邪就自己折腾去吧，人的成长都需要一个过程，等软件公司交了很多学费后自然就明白这个道理了开发人员自己以为是好的未必是真好，只有拿很多实际项目去实战、磨合、优化改进，用很多开发人、同事来反复做试验做测试、用很多客户来当小白鼠做了N多试验后，才能最后能得到成熟稳定的、度可复用的、精品软件组件，只是纸上谈兵，其实是个瞎扯蛋的玩意儿、一拿到实战就更本经不起百折腾。每个软件组件都是需要用钱堆起来的，要么用自己的钱堆起来、要么购买别人用钱堆起来的成组件，例如我们购买“操作系统、数据库、开发环境、第三方的组件”等等都是同样的道理。

有时候想想，就数据库访问组件，有啥了不起的，但是自身的残酷经历告诉我，又能适应B/S、又能应C/S，又能适应小项目，又能适应大项目的高效率数据库访问组件而已，就在这个组件上反反复复知道走了多少弯路，修改修正了多少Bug，经历了前后很多年后，才稳定下来，才敢拿出手，才敢放在项目里用，一方面可能是我水平太差，另一方面我搞出来了世界也变了，微软已经出来更多更好的关解决方法？所以跟这技术屁股后面跑，跑得累死，最后啥也没捞到，甚至同事的认可也没能得到也可能的，你是一个人搞技术，人家是几十、几百个人专业搞技术，你搞的是技术的低端、平时还要折客户的业务系统，所以我也认清了自己，搞好业务、搞好日常管理、搞好客户关系，搞好产品质量才应道理，技术性的东西，搞得越简单越好，越傻瓜越好，越成熟越稳定越好，越是能通过拿来主义的往是越省心省事，实在不行购买也可以，花钱办事嘛，自己也懒得在这个上花费更多精力了，时间不人，机会不等人。 谁愿意去折腾乱八七糟的，就去折腾吧，也懒得管了，过了几年后连别人的认可也能达到，瞎折腾一场，自然就明白了，现在怎么教育也没用，他也不会信那个邪的、而且精力也旺盛很，不要过多的浪费口舌，说不定还可能落个打击人家积极性的帽子。