

何玥13211114

博客园

首页

新闻

新随笔

联系

管理

订阅

随笔 - 2 文章 - 0 评论 - 1

昵称：何玥13211114

园龄：4个月

粉丝：0

关注：0

+加关注

<2017年2月>

日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4
5	6	7	8	9	10	11

搜索

找我看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

我的标签

软件工程(2)

云计算(1)

比较(1)

结构化方法(1)

结合(1)

面向对象方法(1)

随笔档案

2016年12月 (1)

2016年10月 (1)

最新评论

1. Re: 云计算对传统软件工程的影响

我也觉得搞信息安全其实可以在这上面做点事情，@娄老师

--ffl

阅读排行榜

1. 云计算对传统软件工程的影响(55)

2. 软件工程之结构化方法与面向对象方法之比较与结合(34)

评论排行榜

1. 云计算对传统软件工程的影响(1)

软件工程之结构化方法与面向对象方法之比较与结合

软件开发方法指，在项目投资规模和时间限制内，设计、实现符合用户需求的高质量软件，根据软件开发的特点，提出的多种软件开发的策略。随着20世纪60年代，计算机软件、硬件发展不均衡，使大型软件的开发过程中出现了复杂程度高、研制周期长、正确性难以保证的三大难题，引发了“软件危机”。为了同时提高软件效率和质量，软件开发方法不断革新。经过几十年的研究和应用，两种基于相应的程序设计思想和语言的软件开发方法，结构化方法与面向对象方法，成为了主流的开发方法之一，广泛地使用于软件工程。

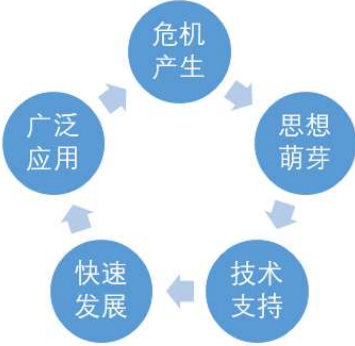
结构化方法包括结构化分析(Structured Analysis，简称SA)、结构化设计(Structured Design，简称SD)和结构化程序设计(Structured Program Design，简称SP)三部分内容。相应地，面向对象方法包括面向对象分析(Object-Oriented Analysis，简称OOA)、面向对象设计(Object—Oriented Design，简称OOD)和面向对象程序语言(Object-Oriented Program Design，简称OOP)。两种软件开发方法从起源、思想、分析、设计，到程序设计、扩展重用、应用等各个方面有着许多的联系和区别，下文我将对二者进行比较分析。两种方法针对不同的工作环境和应用场景，各具优势，也都有所不足，我也将讨论二者在软件工程中的结合，以期产生更好的效果。

(一) 从起源上看

结构化方法与面向对象方法都起源于相应的程序设计思想和语言。20世纪60年代后期，《程序结构理论》和《GOTO陈述有害论》的提出，证明了任何程序的逻辑结构都可以用顺序结构、选择结构和循环结构来表示，确立了结构化程序设计思想，产生了如FORTRAN、PASCAL、C等语言。结构化方法把对程序的分析、设计，延伸至对项目工程的分析、设计，结合程序设计语言的技术支持，得以产生和发展。

20世纪80年代，随着应用系统的日趋复杂、庞大，结构化开发方法在工程应用中出现了一些问题。同期，面向对象程序设计思想经过20年的研究和发展逐渐成熟，一大批面向对象语言相继出现，面向对象方法自产生就广受青睐。90年代中期，互联网兴起，JAVA语言因跨平台特性得以蓬勃发展；21世纪初，不受限于时空的联合开发成为常态；今天，移动APP市场火爆，Andriod开发成为热点。面向对象方法已经成为软件开发方法的中坚力量。

结构化方法和面型对象方法的起源和发展具有模式一致性：



图一：软件开发方法发展模式

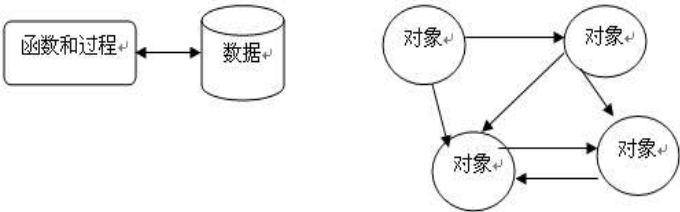
(二) 从思想上看

结构化方法承袭了结构化程序设计的思想，把待解决的问题看作一个系统，用系统科学的思想方法来分析和解决问题。结构化方法遵循抽象原则、分解原则和模块化原则；以数据和功能为中心；以模块为基本单位；以算法为程序核心；强调逐步求精和信息隐藏。

面向对象方法的思想是模拟了客观世界的事物以及事物之间的联系。面向对象以类取代模块为基本单位；通过封装、继承和多态的机制，表征对象的数据和功能、联系和通信；通

通过对对象的管理和对象间的通讯完成信息处理与信息管理的计算和存储，实现软件功能。

对于结构化方法，模块由函数实现，完成对输入数据的加工和计算，数据和功能是分离的；而面向对象把数据和功能封装在对象中，形成一个整体。两种方法在数据和功能上的不同处理是其思想上的本质差别。



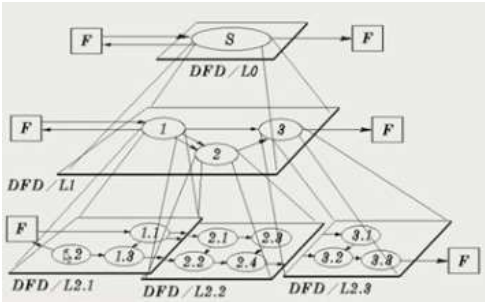
图二：结构化思想和面向对象思想

（三）从分析上看

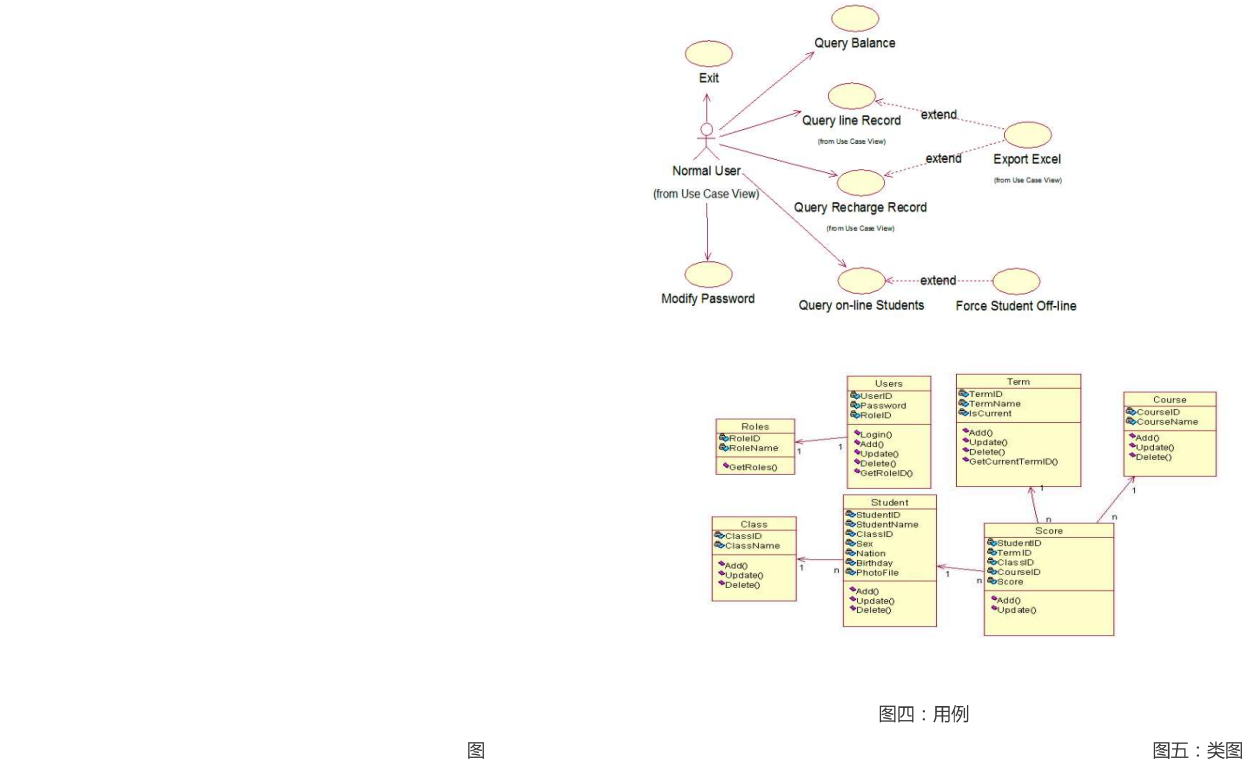
建立模型是为了更好地理解要模拟的现实世界，是软件开发方法的核心问题。在结构化方法中，使用SA构建系统的环境模型和逻辑模型，实现模型的主要工具有数据字典(DD)、ER图和数据流图(DFD)。数据字典是一个包含所有系统数据元素定义的仓库；ER图描述了数据之间的属性及联系；数据流图是描述信息流和数据从输入到输出变换，并展示系统和外部的接口、数据的输入和输出以及数据的存储的应用图形技术[1]。SA的前提条件是需求分析，建模过程是迭代分解需求、不断细化应用的过程，即数据流图的分解从顶层图开始，按照每个加工对应一个子图的分解原则，逐层分解为0层图、1层图等。

面向对象方法使用OOA定义类，对现实世界建模。OOA的主要任务是要在问题域上构建具有主题层、对象层、结构层、属性层和服务层的OOA模型，实现模型的主要工具有用例图(Use-Case)和类图(Class Diagram)。用例图从用户角度描述系统功能，并指出各功能的操作者，是对需求分析的整理；类图定义了类的组成（属性和服务）、类的结构和类间的关系，确定并划分系统中的类。经过OOA，系统的静态模型建立起来。

相对于结构化方法使用DD、ER图和DFD分别描述数据和功能（尽管二者存在相互参考），面向对象方法中，无论是用例图还是类图，数据和功能的描述总是并行的。



图三：数据流图



图四：用例

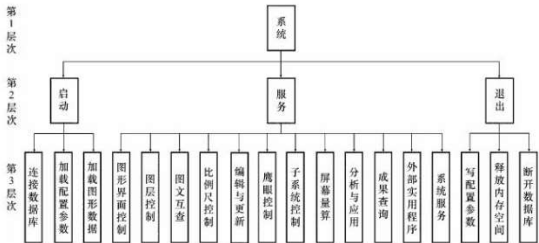
图五：类图

（四）从设计上看

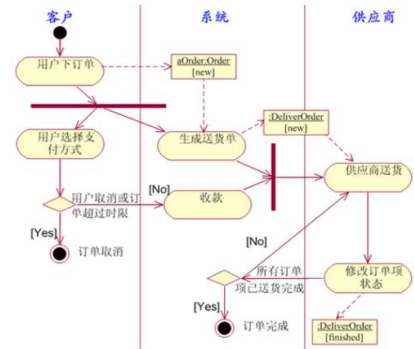
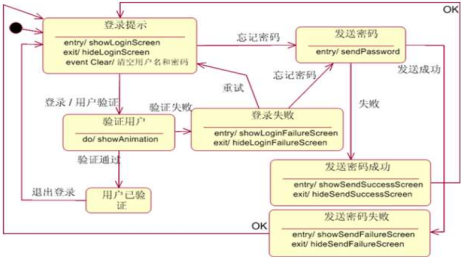
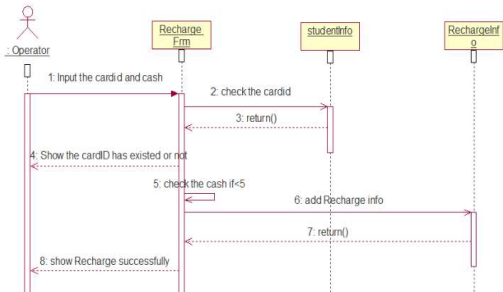
在结构化方法中，使用SD构建系统的行为和功能模型，实现模型的主要工具有模块结构图(SC)和程序流程图。模块结构图说明了系统的模块的划分、模块的功能、模块间的数据传选及调用关系。根据数据流图，我们能够映射出相应的软件的上层模块结构；结合数据流图，我们可以逐步分解上层模块，设计出中、下层模块；对于得到的全部模块，我们需要设计模块基本的内部结构和外部接口及对模块结构进行优化。进一步，则是针对每一个模块设计程序流程图，整理优化，归纳算法。SD依然是对项目系统的进一步分解求精的过程，把模型从逻辑级，细化到模块级，再细化到程序级。

而OOD不只是对OOA的细化，更主要的，构建了系统的动态模型，实现模型的主要工具有交互图(Sequence Diagram)、状态图(State Diagram)、活动图(Activity Diagram)。和模块相比，对象最大的不同是具有“活性”，突出表现在对象具有状态和对象间能够通讯。交互图描述对象间的交互关系，显示对象之间的动态合作关系，强调对象之间消息发送的时间顺序；状态图描述对象的所有可能状态，以及事件发生时状态的转移条件；活动图描述为满足用例要求所进行的活动以及活动间的约束关系，用于识别并行活动，以提高系统效率[2]。

从设计方面，我们可以比较明显地看出结构化和面向对象的区别。结构化方法的核心是程序，从分析到设计，其实是从抽象到具体，从模糊到清晰地实现程序的过程；而面向对象方法的核心是功能，分析的是对象，设计的是行为，程序设计和系统设计相对分离。



图六：模块结构图



图七：交互图

图八：状态

图

图九：活动图

(五) 从程序设计上看

以例子进行说明（银行存取款）：

```
int main () {
    int count;
    int money;
    int type;
    scanf ("%d %d %d", &count, &money, &type);
    if (type)
        printf ("%d", count - money);
    else
        printf ("%d", count + money);
}
```

结构化程序设计

```
class Count {
    int count;

    Count (int count) {
        this.count = count;
    }

    int save (int money) {
        count += money;
    }

    int take (int money) {
        if (count >= money)
```

```
        count -= money;  
    }  
}
```

面向对象程序设计

从上例可见，结构化程序设计是一种过程式的“解题”的方式，程序关注且只关注对于输入数据，输出正确的结果，代码是算法的直接体现，代码效率高；面型对象程序设计是整体式的“建模”的方式，程序关注现实客体，而非某些数据，代码是功能的直接体现，复杂的算法往往是一两行库函数处理，代码效率低。

（六）从扩展重用上看

结构化方法是面向整体应用进行的分析、设计，程序设计也是过程式的针对固定的输入域，代码定向性明显。所以，结构化方法的可扩展性较差，功能的变化可能危及整个系统；重用性不好，若系统间存在嵌套关系，主系统可重用子系统；较难以组合拼接，系统的设计实现是紧耦合的，连接往往是有缝的。

相反，面向对象方法虽然基于应用，但面向现实客体，对象之间独立性较强，功能是对象的交互。所以，面向对象的可扩展性较强，只需扩展或修改某个类，或调整某种通讯；重用性好，面向对象的继承和多态机制大大提高了代码重用的层次和粒度；易于组合拼接，对象是数据和功能的最小单位，为对象建立新的通信的联系，就能够组合出新的软件系统。

（七）从应用上看

结构化方法的实质是问题求解，结构化程序是由算法决定的，算法是程序员分析设计的，程序的执行过程主要是由程序员控制，而不是由用户控制；面向对象方法中，程序员设计的是对象属性及操作方法，但在什么时间、使用什么方式操作对象则是完全由用户交互控制。

结构化方法的建模工具难以表达交互性强的软件系统，程序设计融入系统的分析和设计中，处理大型系统时会过于复杂，甚至很难控制；面向对象方法的抽象机制提供了自然的建模方法，特别是能很好地把握对象之间复杂的相互关系。

结构化方法比较适合工程计算、实时数据的跟踪处理、各种自动控制系统等等；面向对象分析更加适用于复杂的、由用户控制程序执行过程的应用软件，比如大型游戏软件以及各类管理信息系统软件[3]。

（八）二者之结合

经过上述分析，我们可知结构化方法和面向对象方法对于不同的软件系统各有优劣。结构化方法把解空间分数据和功能两部分，可以更加清晰地进行需求分析和功能分解，数据流程图能够细致地说明数据在各个功能模块之间的流动和变化，更适于系统设计的前期阶段。设计人员清楚地了解数据和系统要求的操作后，面向对象方法能够把数据和功能以对象为单位封装成一个整体，更直观地表达对象的状态变化和对象间的交互，更加准确地分析功能的实现过程，更适于在软件后期细化系统的具体行为。基于此，设计的混合式软件开发方法如下：

- 1) 使用SA进行需求分析，建立数据字典，构建总的和分层的数据流图。
- 2) 使用模块结构图设计系统的独立功能块，做出模块内的程序流图。
- 3) 结合数据流图，聚合同类模块，规约类，根据程序流图，设计类的属性和类的方法。
- 4) 使用OOD建立系统的动态模型，分析对象的行为和协作。
- 5) 总体面向对象程序设计，细节结构化程序设计优化，实现代码层。

使用混合式方法，我们能够充分利用两种方法的优点，扬长避短，提高开发的效果和效率。

无论是结构化方法，还是面向对象方法，都是用来解决日益矛盾的软件危机的系统方法。从直接开发，到结构化方法，再到面向对象方法，软件构件的愈发独立、可重用，开发在一个更高的层次进行，分析层、设计层和代码层关联性减少。这些都有利于系统开发员更加关注功能本身，提高软件质量。硬件性能的提高会使计算机的使用越发广泛，软件工作的环境更加复杂，软件的功能更加丰富，软件的性能更需提高，对软件开发方法提出了更多的

要求，会涌现更高层次的新的方法。无论使用哪种开发方法，或者是混合哪几种开发方法，我们都要因地制宜，依据需求分析和系统要求，做出最好的选择或组合。

参考文献：

[1] 汪寒昊，谢加胜，邢跃，《软件开发方法——结构化方法与面向对象方法比较》，绵阳师范学院数学与计算机科学学院，2011，第13期。

[2] 张莉，《结构化方法与面向对象方法的比较分析》，陕西师范大学计算机科学学院，陕西师范大学学报（自然科学版），第29卷，第2期，2001.6。

[3] 许秀林，《结构化方法与面向对象方法思想辨析》，南通职业大学，南通职业大学学报，第18卷，第4期，2004.12。

标签：[结构化方法](#)，[面向对象方法](#)，[软件工程](#)，[比较](#)，[结合](#)

好文要顶

关注我

收藏该文

[何玥13211114](#)
关注 - 0
粉丝 - 0
[+加关注](#)

0

0

« 上一篇：[云计算对传统软件工程的影响](#)

posted @ 2016-12-18 00:07 何玥13211114 阅读(34) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【秘密】藏在春节红包里的技术秘密
- 【推荐】融云即时通讯云一豆果美食、Faceu等亿级APP都在用
- 【活动】一元专享1500元微软智能云Azure



- 最新IT新闻：
- 2018年起，YouTube将不再支持30秒强制广告，你怎么看？
 - 尴尬的中国手机一哥争夺战：越卖越多 却沦为渠道的打工者
 - 变现太难成本太高 16WiFi被迫关停十几个城市业务
 - 新材料让钠离子电池寿命可媲美锂电池
 - 亚马逊死磕沃尔玛：也提供超35美元包邮
- » 更多新闻...



- 最新知识库文章：
- 「代码家」的学习过程和学习经验分享
 - 写给未来的程序媛
 - 高质量的工程代码为什么难写
 - 循序渐进地代码重构
 - 技术的正宗与野路子
- » 更多知识库文章...

