

一、 Object-oriented design OO 设计.....	2
二、 Modular decomposition 模块分解.....	3
三、 Outside-in design 由外而内的设计.....	4
四、 设计方法比较.....	5

一、Object-oriented design OO 设计

- 1 确定对象的类和它们之间的相互关系
- 2 然后描述对象的属性和方法，解释对象之间的关系

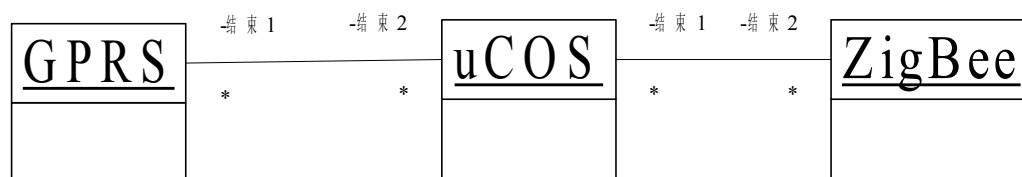


图 1-1 I 级视图

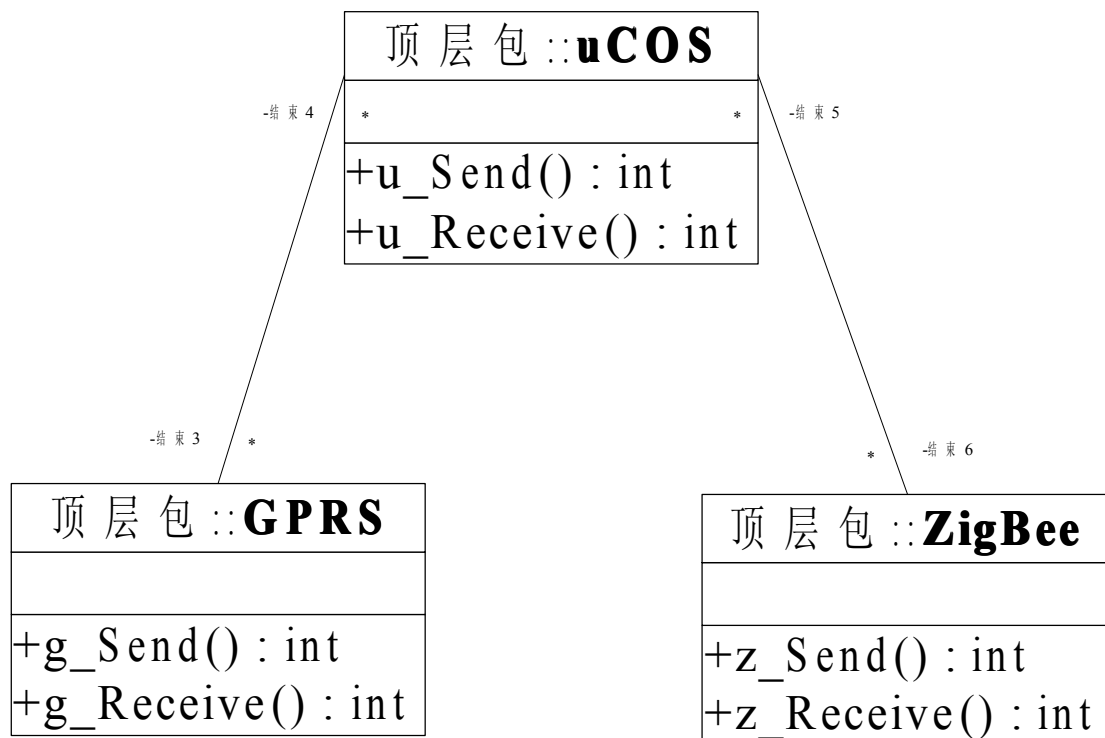


图 1-2 II 级视图

二、Modular decomposition 模块分解

高层的功能描述->将功能分配给组件->每一个低层组件是如何组织起来的及组件之间的关系。

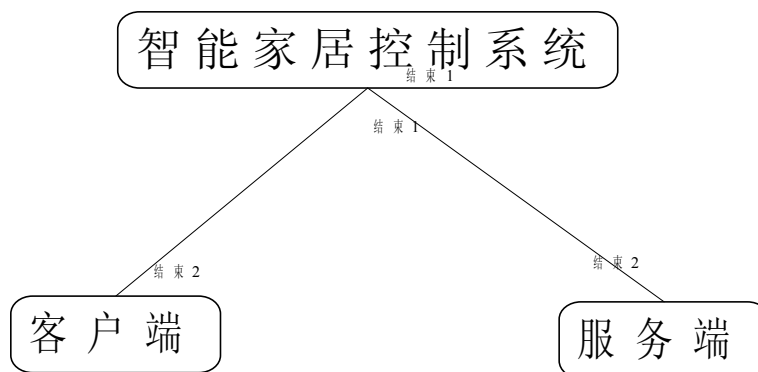


图 2-1 I 级视图

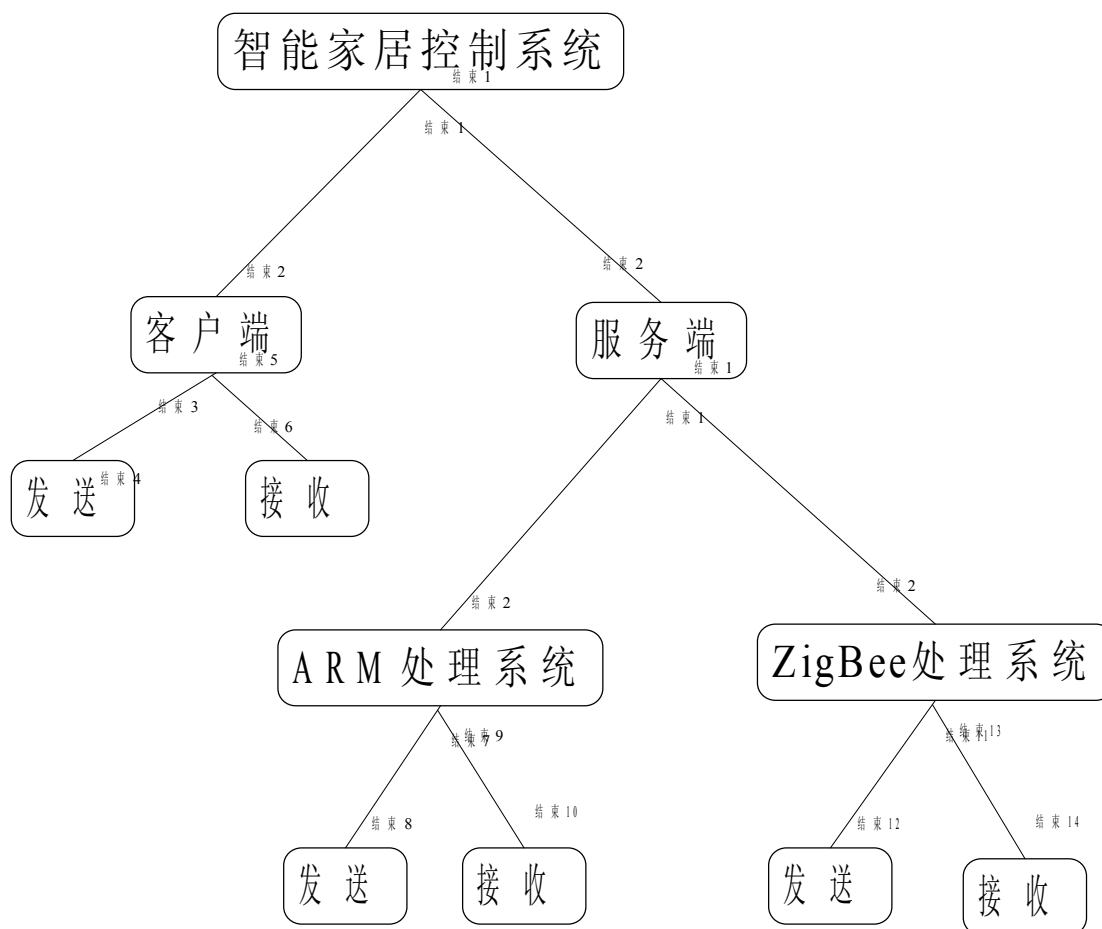


图 2-2 II 级视图

三、Outside-in design 由外而内的设计

基于系统的用户输入（属黑盒方法），即系统如何处理每一个输入及产生什么样的输出。

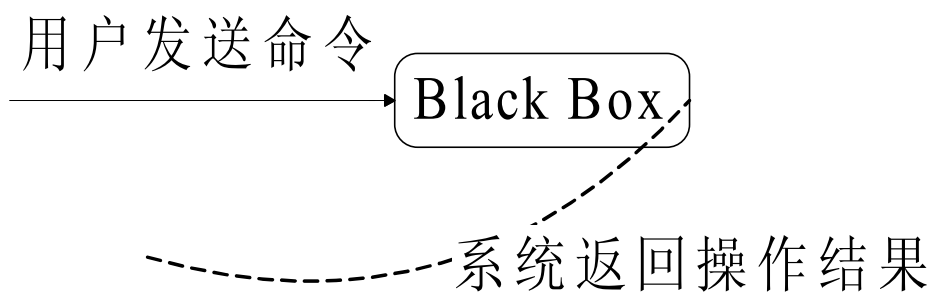


图 3-1 I 级视图

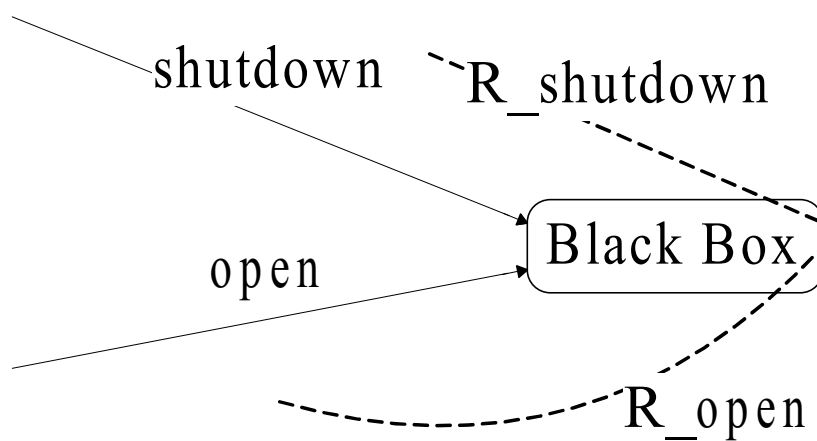


图 3-2 II 级视图

四、设计方法比较

4.1 Object-oriented design (OO 设计) 面向对象设计

优点：便于我们思考和建立模型。我们自然而然就能设计出整个系统的模型，而且规范，技术成熟，便于控制和管理。

缺点：如果我们并不需要关注其全部细节，或者我们的着眼点并不放在对象上，比如重点的设计是在过程上，这个方法会稍显繁杂冗余，或者是在不必要详细的地方投入了过多的经历。

4.2 Modular decomposition 模块分解

优点：我们主要把精力集中在功能处理的各个模块上了，在模块操作的内部能进行详细的描述，便于指导程序员的开发。另外专门针对制定的功能，更便于开发人员了解待开发的的功能的需求。

缺点：对于所处理的对象和数据的表现力不足，也无法完整地体现出程序的组成的逻辑整体结构。

4.3 Outside-in design 由外而内的设计

优点：表现了用户的输入和产出，直截了当，我们并不关心内部如何设计，只关心其功能，验证相应输入是否得到相应输出。这样对 黑盒 测试是非常方便的。我们也可以细化的每个模块，分别对每个模块进行测试。

缺点：对系统的内部和实现描述几乎为 0，对我们了解整个系统起不到任何帮助。