

软件质量保证与测试

Software Quality Assurance and Testing

第 5 章 集成测试

5.3 自顶向下和自底向上集成



金陵科技学院

5.3 自顶向下和自底向上集成

在实际的软件测试工作中，增量式集成是较为普遍采用的，增量式集成又可以分为自顶向下和自底向上两种典型的情况。

5.3.1 自顶向下增量式集成

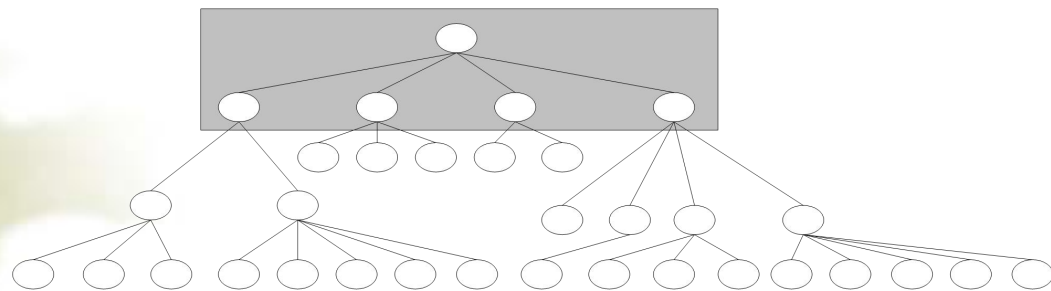
自顶向下增量式集成是指，依据程序结构图，从顶层开始，按照层次由上到下的顺序逐步扩大集成的范围，增加集成的模块，来进行集成测试。

在逐步扩大集成范围，增加集成模块的具体路径选择上又可以分为广度优先和深度优先。

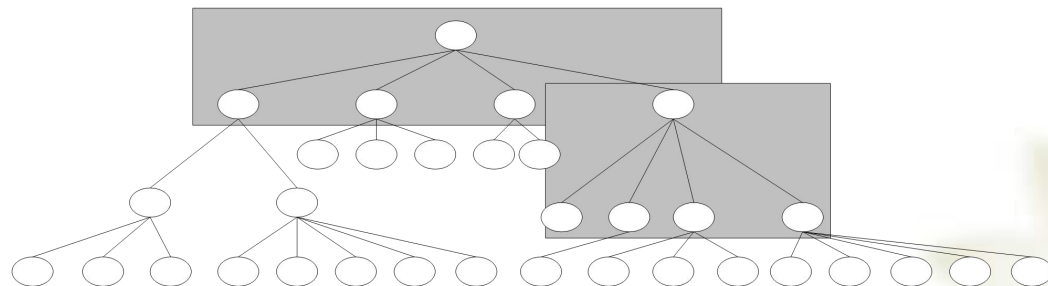
5.3.1 自顶向下增量式集成

具体步骤：

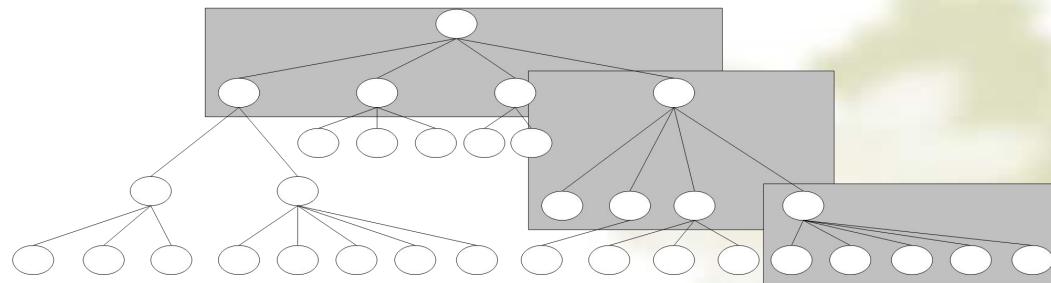
- ① 从软件结构图的树根开始，将程序的主控模块作为测试驱动；
 - ② 根据集成的路径（深度优先或广度有限），每次加入一个或者几个已经完成了单元测试的下级模块，其它相关模块均用桩模块代替；
 - ③ 进行集成测试，测试集成新模块后有没有产生错误；
 - ④ 上述过程从第2步重复进行，直到所有都集成测试完毕。
- 以深度优先为例，自顶向下增量式集成的过程如后图所示。



顶层子树（会话1-4）



第二层子树（会话12-15）



底层子树（会话38-42）

5.3.1 自顶向下增量式集成

自顶向下增量式集成的优点：

- ① 集成测试可以较早的开始
- ② 减少了驱动模块的开发
- ③ 如果底层接口未定义或可能修改，则可以避免提交不稳定的接口

自顶向下增量式集成的缺点：

- ① 桩模块的开发代价较大
- ② 底层模块的无法预料的条件要求可能迫使顶层构件的修改
- ③ 在软件集成后，对底层模块的调用和测试不够充分。

5.3.2 自底向上增量式集成

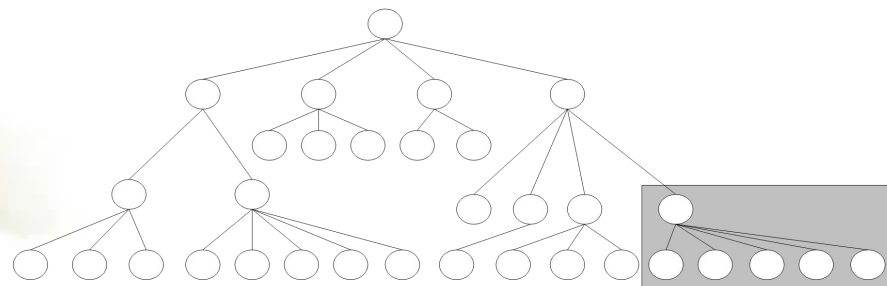
底向上增量式集成是指，依据程序结构图，集成从最底层的模块开始，按照层次由下到上的顺序逐步扩大集成的范围，增加集成的模块，来进行集成测试。

同样，在逐步扩大集成范围，增加集成模块的具体路径选择上也可以分为广度优先和深度优先。

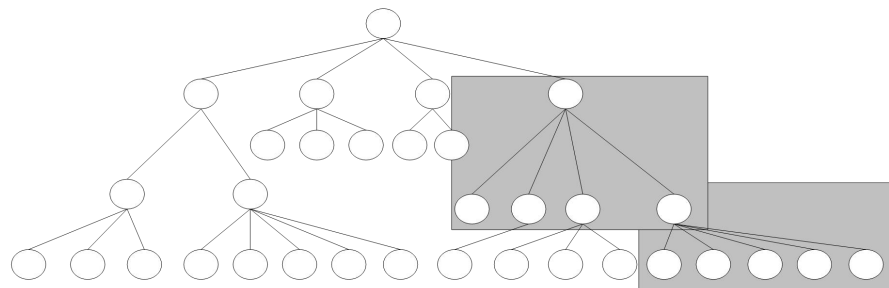
5.3.2 自底向上增量式集成

具体步骤：

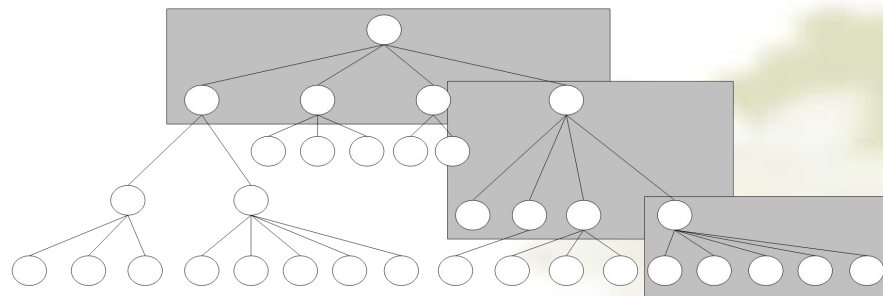
- ① 从软件结构图的叶子节点开始，逐步增加上级或者同级模块；
 - ② 根据集成的路径（深度优先或广度有限），每次加入一个或者几个已经完成了单元测试的上级或者同级模块，其它相关模块均用驱动模块代替；
 - ③ 进行集成测试，测试集成新模块后有没有产生错误；
 - ④ 上述过程从第2步重复进行，直到所有都集成测试完毕。
- 以深度优先为例，自顶向下增量式集成的过程如后图所示。



底层子树（会话13-17）



第二层子树（会话25-28）



底层子树（会话29-31）

5.3.2 自底向上增量式集成

自底向上的增量式集成（Bottom-Up Integration）方式是最常使用的方法。这种方式从最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块（包括子模块的所有下属模块）事前已经完成组装并经过测试，所以不再需要编制桩模块。

5.3.2 自底向上增量式集成

自底向上的增量式集成优点如下：

- ① 多组底层叶节点的测试和集成可以并行进行
- ② 不限制可测试性，对底层模块的调用和测试较为分
- ③ 实现方便，不需要桩模块
- ④ 测试人员能较好地锁定软件故障所在位置

5.3.2 自底向上增量式集成

自底向上的增量式集成缺点如下：

- ① 驱动模块的开发成本较大
- ② 高层构件的可操作性和互操作性测试得不够充分
- ③ 对于某些开发模式不适用，如使用XP开发方法，它会要求测试人员在全部软件单元实现之前完成核心软件部件的集成测试。

5.3.2 自底向上增量式集成

自底向上的增量式集成适用情况

- ① 实现具体功能的复杂代码在底层（多数软件都是如此）
- ② 在子系统的迭代和增量开发中，支持单位范围内的测试
- ③ 重要构件在底层的系统

自底向上的集成测试方案是工程实践中最常用的集成测试方案。相关技术也较为成熟。

5.3.3 两种增量式集成的对比

	自顶向下集成	自底向上集成
优点	减少了驱动模块的开发；一开始便能让测试者看到系统的框架可以自然地做到逐步求精；如果底层接口未定义或可能修改，则可以避免提交不稳定的接口。	多组底层叶节点的测试和集成可以并行进行；不限制可测试性，对底层模块的调用和测试较为充分；实现方便，不需要桩模块；测试人员能较好地锁定软件故障所在位置；由于驱动模块模拟了所有调用参数，即使数据流并未构成有向的非环状图，生成测试数据也没有困难；特别适合于关键模块在结构图底部的情况。

5.3.3 两种增量式集成的对比

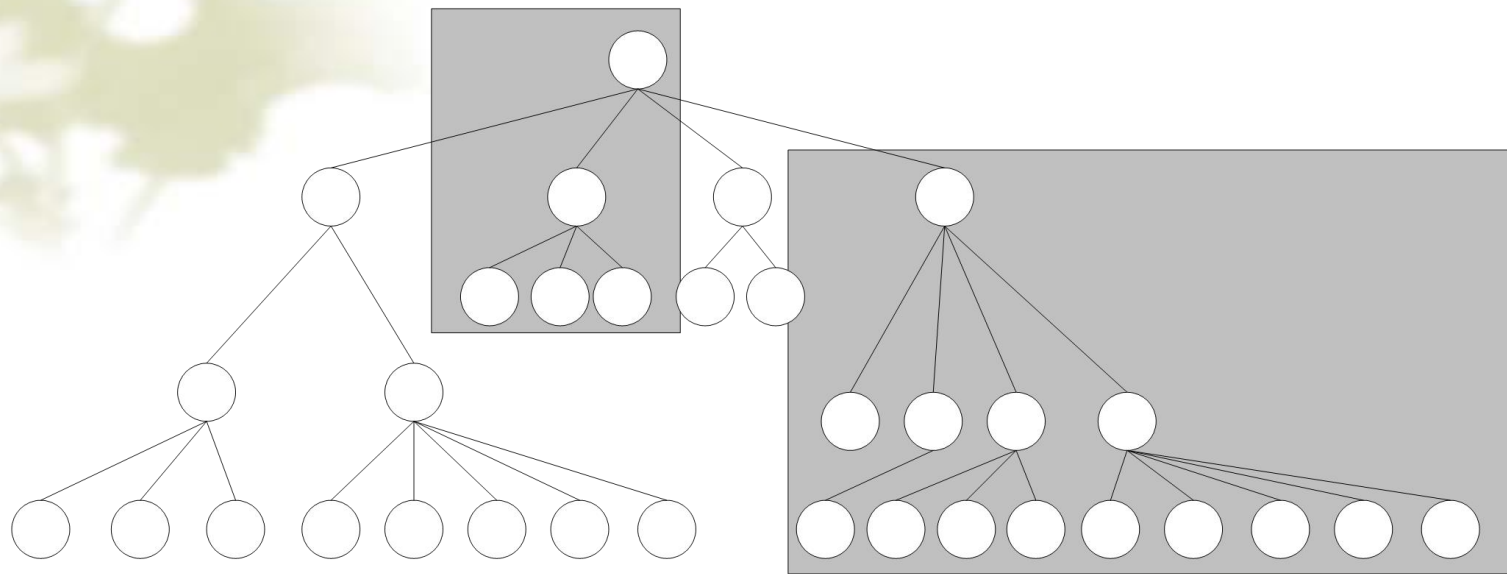
	自顶向下集成	自底向上集成
缺点	<p>桩模块的开发代价较大；</p> <p>底层模块的无法预料的条件要求可能迫使上层模块的修改；</p> <p>在软件集成后，对底层模块的调用和测试不够充分；</p> <p>在输入/输出模块接入系统以前，在桩模块中表示测试数据有一定困难；</p> <p>由于桩模块不能模拟数据，如果模块间的数据流不能构成有向的非环状图，一些模块的测试数据难于生成；</p> <p>观察和解释测试输出往往也是困难的。</p>	<p>需要驱动模块；</p> <p>高层构件的可操作性和互操作性测试得不够充分；</p> <p>对于某些开发模式不适用，如使用XP开发方法，它会要求测试人员在全部软件单元实现之前完成核心软件部件的集成测试；</p> <p>整个程序（系统）的框架要后期才能看到；</p> <p>只有到测试过程的后期才能发现时序问题和资源竞争问题。</p>

5.3.4 三明治式集成

自顶向下集成和自底向上集成各有其优缺点，为了取长补短，可以把两者结合起来使用，这就是三明治式集成。

三明治式集成的过程如后图所示。

5.3.4 三明治式集成



三明治式集成把自顶向下和自底向上集成结合起来使用后，可以同时具有自顶向下和自底向上两种集成方式的一些优点之外，能够减少一些桩模块和驱动模块的开发。

本节内容就讲到这里，谢谢，再见！



金陵科技学院