

# Class Design of Process Planning in MES for Semiconductor Final Test

Ren Wang<sup>1,2</sup>, Hua Ye<sup>1,2</sup>, Haobo Liu<sup>1,2</sup>

<sup>1</sup>School of Automation, Southeast University, Nanjing Jiangsu

<sup>2</sup>Key Laboratory of Measurement and Control of Complex Systems of Engineering of Ministry of Education, Southeast University, Nanjing Jiangsu

Email: wangren725@163.com

Received: Nov. 2<sup>nd</sup>, 2016; accepted: Nov. 21<sup>st</sup>, 2016; published: Nov. 24<sup>th</sup>, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

MES has been widely used in Semiconductor test manufacturers for increasing productivity. The function of process planning has been required because of varied requirements to test process and procedures from customers. Additionally, classes of procedures should be able to be extended easily in MES by developer when it's necessary. We designed the entity classes according to the requirements of test process and each test procedure, and extendibility has been achieved by applying composite pattern. Test results show that process can be customized and test data can be saved properly, the existed class formation can be extended easily as well. The developed function can be used in MES for final test properly.

## Keywords

Semiconductor Final Test, MES, Process Planning Composite Pattern

---

# 半导体FT测试MES的流程定制功能的类设计

王仁<sup>1,2</sup>, 叶桦<sup>1,2</sup>, 刘浩波<sup>1,2</sup>

<sup>1</sup>东南大学自动化学院, 江苏 南京

<sup>2</sup>东南大学复杂工程系统测量与控制教育部重点实验室, 江苏 南京

Email: wangren725@163.com

## 摘要

使用MES提高生产力已经成为半导体测试厂商的常见手段，而客户对FT测试流程及测试步骤的多样化需求要求MES软件必须具有灵活定制测试流程的功能，并在必要时能够由开发人员在MES软件中快速扩展出新种类的测试步骤。本文根据FT测试流程及其中各类测试步骤的业务特点进行了类设计，并使用“组合”设计模式使其具有优良的可扩展性。测试结果表明，基于该类设计的功能实现满足了流程测试步骤灵活制定、测试结果数据保存等功能，并具有十分优良的可扩展性，可较好的应用于半导体FT测试MES软件中。

## 关键词

半导体FT测试，MES，流程定制组合设计模式

## 1. 引言

半导体 FT 测试(Final Test, 也称为 FT)是对已制造完成的半导体元件进行结构及电气功能确认，以保证半导体元件符合系统的需求[1]。半导体行业是典型的技术密集型科技制造行业，与信息化的企业级应用软件相结合对提高生产效率至关重要，而其复杂多变的测试流程则对应用软件的设计提出了更高的要求。

制造执行系统 MES (Manufacture Execution System)是一种以优化整个生产过程为目标，实时收集和 处理生产过程中的数据，同时与计划层和控制层保持双向通信并从上、下两层接收相应数据、反馈处理 结果和生产指令的企业级应用软件[2]。MES 是半导体测试行业常见的应用软件，自上世纪八十年代初在 美国企业界兴起后已经成为制造业信息化中成熟的系统。MES 为控制包括物料、设备、人员、流程指令 和设施在内的所有工厂资源，提供了一个统一的平台，对提高企业生产效率、提高产品质量、降低消耗 方面具有重要的作用[3]。根据客户的需求制定灵活可变的测试流程是半导体测试 MES 的核心功能之一， 一些 MES 设计者将其称之为 PRP (Process Planning)模块[1]。

不同的芯片生产者对芯片的测试流程及测试步骤有不同的要求，因此测试厂商的 MES 应当能够根据 芯片生产者的要求在软件系统中建立组装出一条完整的测试流程。在这条测试流程中应当包含客户要求 的各个测试步骤及顺序，而各个测试步骤中则应当包含本测试步骤的测试结果以及特定的业务逻辑。如 如何进行类设计满足上述需求并在必要时能够快速开发扩展出新类别的测试步骤成为 MES 软件的设计中 需要重点解决的问题。

## 2. 功能需求

### 2.1. 行业术语定义

为描述方便，特对行业术语解释如下：

- 来料：指从客户处得到的、未经过测试的封装后芯片；
- 产品：指经过测试、可以发货返还给客户的芯片；
- 生产：指质量检测过程，使得来料转变为产品。
- PE: Process Engineer。流程工程师，负责在系统中为产品制定对应的测试流程。

- OP: Operator. 操作员, 负责在测试生产线上工作, 并同步在 MES 中维护相应信息, 如保存测试数据、继续下一步测试等。

## 2.2. 测试流程及测试步骤

一条完整的 FT 测试流程由许多个测试步骤组成, 每个测试步骤应当记录其自身的测试结果数据并具有特定的业务逻辑(如在何种情况下需要继续、暂停测试等等)。有一类特殊的测试步骤称为测试制程, 一个测试制程可以包括一个或多个子步骤, 是 FT 测试的核心步骤。

FT 测试的测试步骤有如下 9 类:

- IQC (Incoming Quality Control): 来料质量控制, 意为对刚刚进入生产线的来料进行质量检测, 一般为整个流程的第一个步骤。
- Baking: 烘烤, 意为对来料进行烘烤处理, 为接下来的测试步骤做准备。
- GuTest: 机台测试, 令机台对标准件进行检测, 从而校准机台设置。
- Test: 测试制程, 使用测试机台对来料进行检测, 是 FT 测试的核心步骤, 可包括多个子测试步骤, 子测试步骤统称为 FT。测试制程执行后来料转变为未包装的产品。
- Finish: 测试完成站, 一般在该步骤对测试后的产品进行装盘。
- FVI (Final Visual Inspection): 最终光检, 对产品进行可视化检查。
- FQC (Final Quality Control): 最终质量控制, 在装箱前进行最后检查。
- Packing: 装箱, 对装盘完成的产品进行装箱。
- OQC (Output Quality Control): 出货品质控制, 对装箱后产品进行出货前的检查。

以上 9 类测试步骤是目前业内较为通用的测试步骤, 大多数客户对测试流程的要求可通过对以上各类测试步骤进行重复、排列、组合而建立起来。值得说明的是, 半导体行业还处于高速发展阶段, 以上提到的 9 类测试步骤必然无法长期满足日益增长的客户需求, 因此能够允许开发人员快速扩展出新的测试步骤类也是重要需求之一。

## 2.3. 流程定制功能

对于每一批来料, 首先由 PE 针对客户的要求进行流程定制, 再由生产线上的 OP 根据已制定的测试流程依次执行测试步骤对来料进行测试, 如图 1 所示。

例如, 某客户向 FT 测试厂商寄送出一批来料, 并要求对该批来料依次进行如下步骤的测试: IQC- > Baking- > Test(FT1- > FT2)- > Finish- > OQC。其中, Test 为测试制程, 包含两个子测试步骤: FT1、FT2。当测试厂商收到客户的要求后, PE 就会根据客户的要求在 MES 中建立一套包含以上所有测试步骤的测试流程记录表, 生产线上的 OP 就会根据此表依次执行各个步骤, 并将测试结果记录在表中, 示意图如图 2 所示。当有另外一个客户向测试厂商提出不同的测试流程要求时, PE 会根据客户要求, 在 MES 中建立不同的流程步骤与流程记录表, 因此 OP 也会执行不同的操作步骤, 得出不同的测试结果数据。多样化的客户需求要求 MES 必须具有灵活定制测试流程的功能, 以及快速开发扩展的能力。

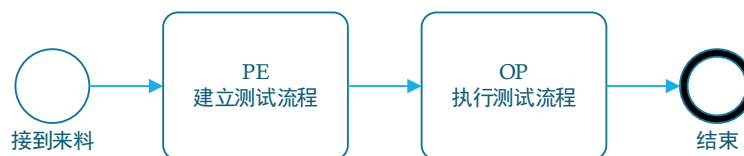
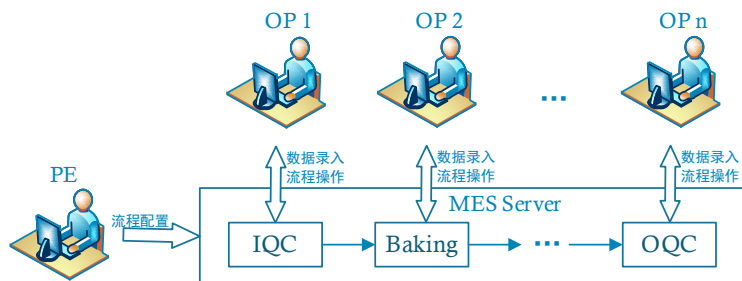


Figure 1. Flowchart: process planning  
图 1. 流程定制业务流程图



**Figure 2.** An example of process planning business  
**图 2.** 流程定制业务举例示意图

对于 PE，MES 流程定制功能应满足以下需求：

1. 可选择一个或多个测试步骤加入到测试流程中；
2. 同一类测试步骤可重复添加；
3. 可利用子测试步骤组装出新的测试制程，并可选的加入至测试流程；
4. 可记录流程中的测试步骤顺序，对顺序敏感；

对于 OP，应满足以下需求：

1. 可在测试过程中对每个步骤的测试结果数据进行录入和保存；
2. 可执行开始步骤、结束步骤、暂停、返工等相关流程操作。

对于开发人员，类设计应当符合“开放 - 封闭原则”[4]，具有优良的扩展性，能够在必要时灵活扩展出新的测试步骤类。

### 3. 抽象建模与类设计

#### 3.1. UML

UML (Unified Modeling Language, 统一建模语言)是面向对象软件的标准化建模语言[5]。UML 因其简单、统一的特点，而且能表达软件设计中的动态和静态信息，目前已成为可视化建模语言的工业标准。本文将使用 UML 对抽象建模和类设计的过程进行描述。

#### 3.2. 基于传统方法的抽象建模与类设计

##### 3.2.1. 抽象建模

小节 1.3 中所描述的流程定制功能需求中主要涉及了三个对象：测试流程、测试步骤以及包含在测试制程中的子测试步骤。首先尝试以传统的树形结构对上述三个对象进行抽象，如图 3 所示。其中，Process 对应测试流程、Node 对应普通测试步骤、ComplexNode 对应特殊的测试步骤——测试制程、Test 对应测试制程中的子测试步骤；Node 与 ComplexNode 为 Process 的叶子节点、Test 为 ComplexNode 的叶子节点。通过建立这样的树形结构可完整的描述出不同的测试流程。

##### 3.2.2. 流程类与测试步骤基类的设计

由于 1.2 小节中所述的 9 类测试步骤拥有共同的属性与操作，因此可抽象出一个基类与流程类建立联系。根据抽象模型以及功能需求可得出流程类与测试步骤基类的设计如图 4 所示。

其中，Process 为测试流程类，包含两个成员变量以及三个方法：lot 变量用于与来料批次绑定；nodes 变量是 Node 对象的列表，用于保存当前测试流程所包括的测试步骤及其顺序信息；以 String 为参数的构造方法通过解析包含流程信息的字符串创建 Process 对象；endCurrentNode()方法用于结束当前测试步骤，

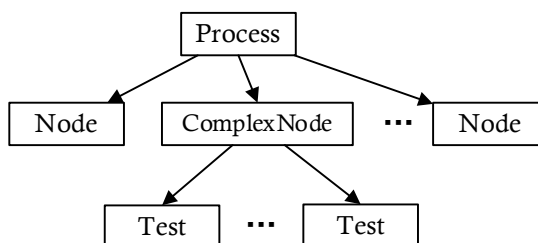


Figure 3. Abstract model based on tree structure  
图 3. 基于“树形”结构的抽象模型

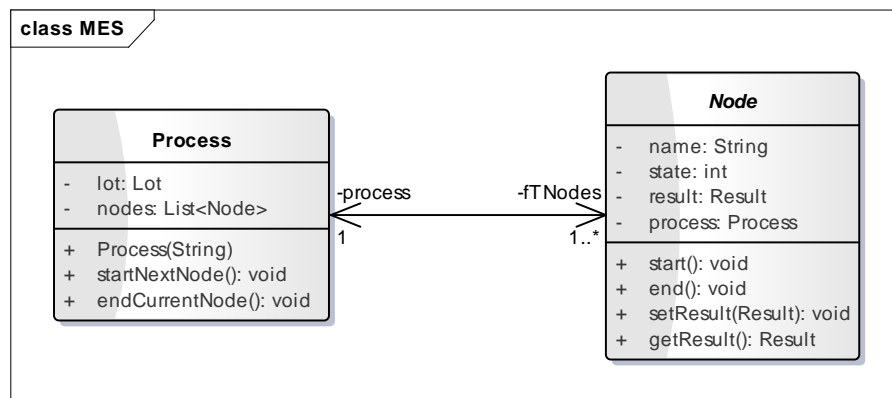


Figure 4. Class of process and base class of test step  
图 4. 流程类与测试步骤基类

即首先找到当前正在进行的测试步骤 Node 类型实例，然后执行该实例的 end()方法；startNextNode()方法用于进入下一个测试步骤，该方法中首先执行 endCurrentNode()方法，然后根据步骤列表寻找到下一个步骤的 Node 类型实例，并执行该实例的 start()方法。

Node 为测试步骤基类，包含四个成员变量以及四个方法：name 变量为当前测试步骤的名称；state 变量为当前测试步骤的状态(未开始、已开始、已结束等)；result 变量为 Result 类型，用于保存当前测试步骤的测试结果数据(如各个 Bin 类所包含的数量等)；start()方法用于进入当前步骤；end()方法用于结束当前步骤；setResult(Result)、getResult()方法分别用于设置、获取当前步骤的测试结果，是成员变量 result 的普通 get、set 方法。

### 3.2.3. 测试步骤子类及子测试步骤的设计

测试步骤除了共同的属性与操作，还拥有自身特有的属性。如 IQC 有重量、标记等特殊属性，Baking 有烤箱温度、烤箱编号等属性需要存储。因此，各类测试步骤均继承 Node 基类并建立自己的子类。详细类设计如图 5 所示：IQC 类为 IQC 测试步骤的抽象，成员变量 mark、netWeight 存储了重量与标记数据；Baking 类为 Baking 测试步骤的抽象，成员变量 ovenNumber 与 timeLimit 记录了烤箱编号以及烘烤时间限制数据；GuTest 类为 GuTest 测试步骤的抽象，由于测试结果的格式不确定，以字符串变量 record 的形式记录了该测试步骤的测试结果；测试制程类命名为 ComplexTest，并持有子测试步骤类 Test 的列表，用于记录子测试步骤的信息；Test 类为子测试步骤 Test 的抽象，成员变量 name、note、result、state、turn 分别记录了步骤名称、备注、测试结果、测试状态、步骤顺序等信息；FVI、FQC、OQC 较为相似，统一抽象为 PassNode 类，以字符串的形式记录结果数据；Packing 与 Finish 相似，统一抽象为 Finish 类，其中 createReelcodes()方法为创建 Reel 盘实例的方法，由于涉及到本文范围以外的类，不再赘述。

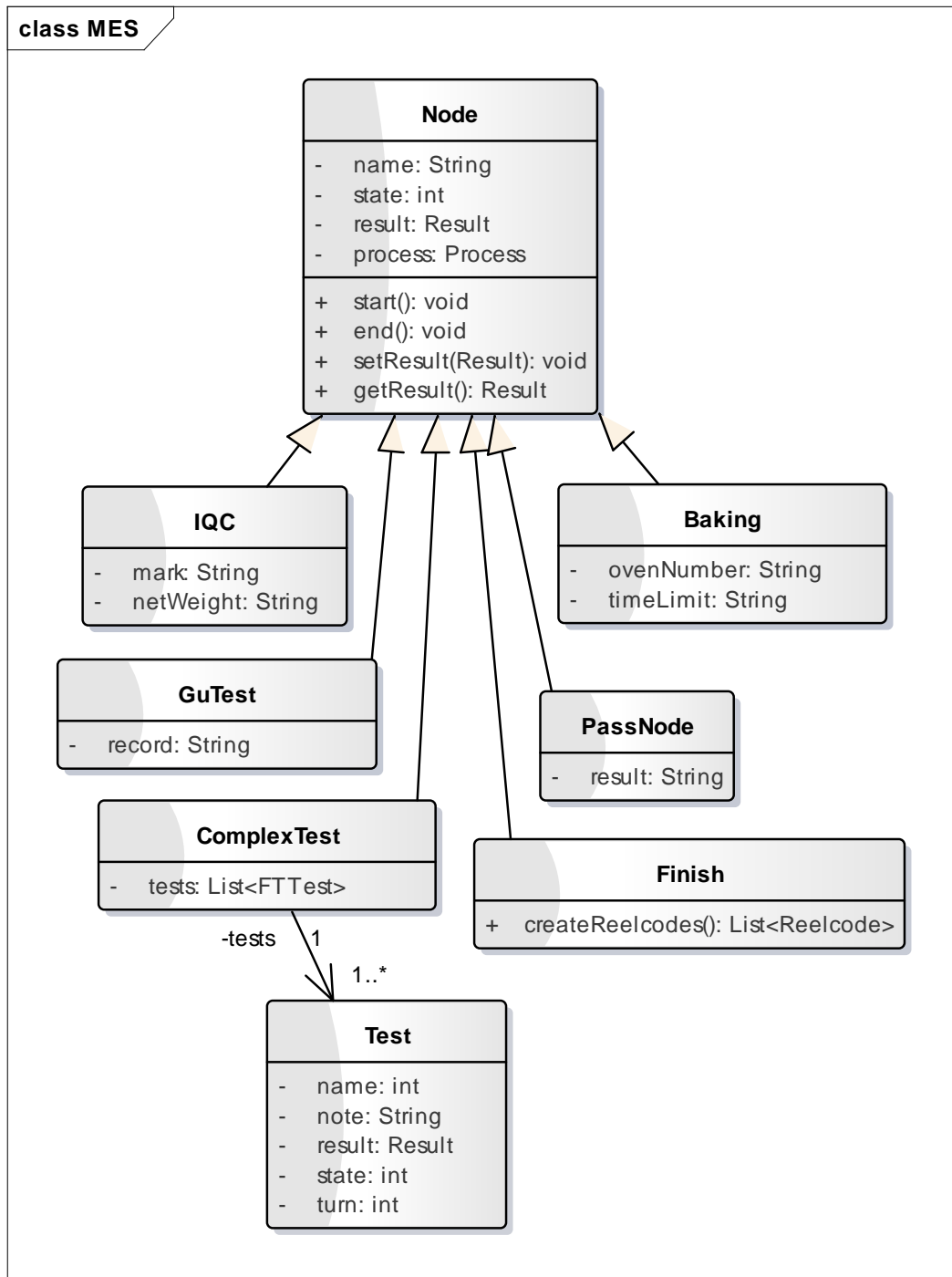


Figure 5. Base class of test step and derived classes

图 5. 测试步骤基类与各测试步骤子类

如上所述的设计虽然可以满足基本的功能需求，但却不具备灵活的扩展性。例如：要建立单独的 Test 子测试步骤，就必须先建立 ComplexTest 测试步骤；要在 ComplexTest 测试步骤中添加 IQC 等步骤，或者要求 ComplexTest 测试步骤中再包含其他测试步骤，就必须修改 ComplexTest 类代码。上述的一系列问题均可通过“组合”设计模式来解决。



### 3.3. 基于组合设计模式的改进类设计

#### 3.3.1. “组合”设计模式

设计模式是一种可以被反复使用，经过分类编目的代码设计经验的总结。使用设计模式可以保证代码的可靠性，提高代码的可维护性，使其可重复利用[6]。

组合模式允许将对象组合成树形结构来表现“整体/部分”层次结构。组合能让客户以一致的方式处理个别对象以及对象组合。组合模式的类图如图6所示。其中，Component类为抽象类，Leaf类与Composite类均继承Component类；Leaf类是Component类的叶子节点(即单一元件)，Composite类是Component类的根节点(即复合元件)。Composite类回旋转持有Component对象列表，将其作为自身的子元件，即可方便的实现树形结构[7]。

#### 3.3.2. 抽象建模

基于2.2.1小节所述的基本树形结构，在保持测试流程与测试步骤的一对多关系、测试制程与子测试步骤的一对多关系保持不变的同时，增加两个抽象对象：“叶子(Leaf)”与“组合(Composite)”。其中非测试制程的测试步骤与子测试步骤在树状结构中为叶子节点，因此归属于“叶子”；测试制程由于可以包含更多的子测试步骤，是根节点，因此归属于“组合”。以此为基础建立模型并应用组合设计模式。

#### 3.3.3. 改进的测试步骤类设计

将组合模式应用至测试步骤类的设计中，得到的类设计如图7所示。其中，将Node设计为抽象基类，并创建LeafNode与CompositeNode两个抽象子类。Node抽象基类作为CompositeNode类的元件，使得CompositeNode类可持有Node基类所有的子类，包括所有叶子节点和根节点，ComplexTest类即为测试制程类。这样的类设计与2.2小节中所描述的类设计的显著不同在于：2.2小节中基于传统方法的类设计只能构建出既定的三层树状结构，而本小节中应用组合设计模式的类设计可通过CompositeNode类的不断嵌套构造出无穷多层的树状结构，极大的提高了可扩展性。

图5中所有原本直接继承Node基类的测试步骤类更改为继承LeafNode类，如图8所示。需要说明的是，原本孤立的Test类现在继承了LeafNode基类，因此既可以作为独立的测试步骤被编排入测试流程，也可以作为子测试步骤编排入ComplexTest测试制程。同样，IQC、Baking等步骤也可以方便的组成新的测试步骤，大大提高了类设计的可扩展性。

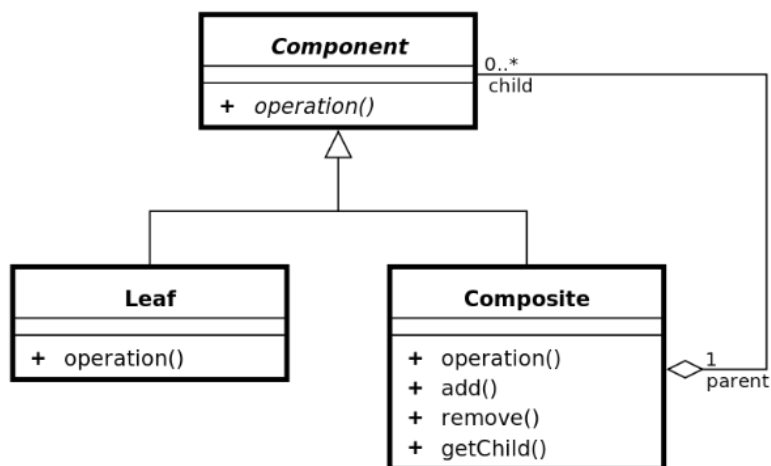


Figure 6. Class diagram of composite pattern

图6. 组合模式类图

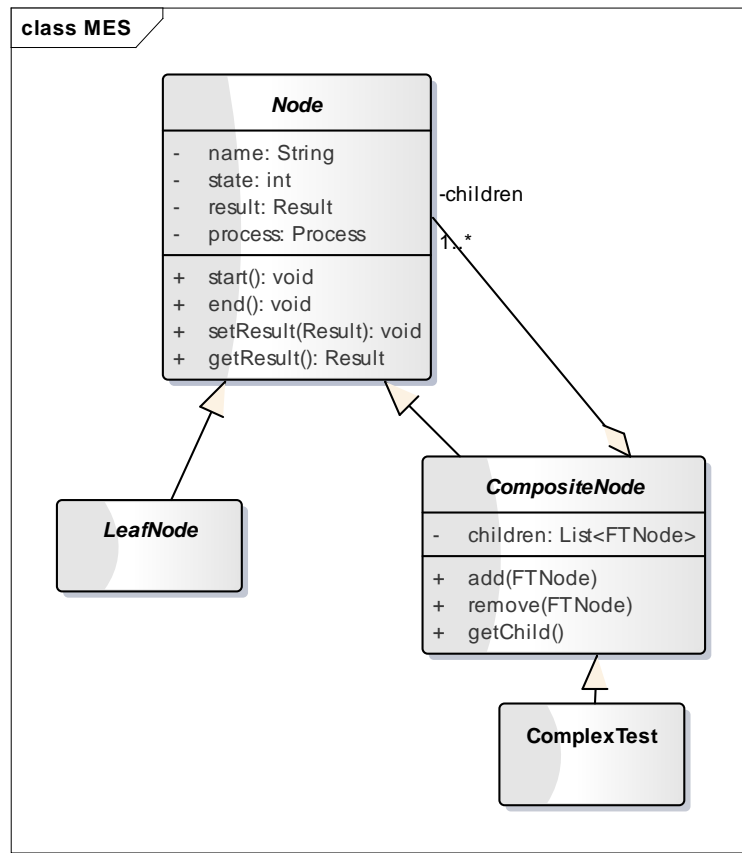


Figure 7. Application of composite pattern on test step classes  
图 7. 测试步骤类中的组合模式应用

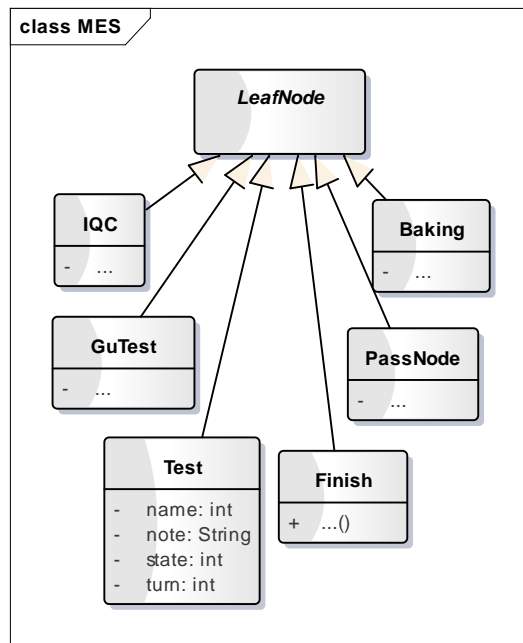


Figure 8. Classes derived from LeafNode class  
图 8. 继承 LeafNode 类的测试步骤类



## 4. 功能运行结果

系统功能采用 B/S 架构，使用 Java Web 相关技术[8]，以 Spring、JPA 作为后台开发框架，以 MySQL 作为数据库具体实现。开发完成后通过 JDK 工具生成 war 包，部署在服务器的 tomcat 容器中运行。

流程组装界面如图 9 所示。其中流程明细一栏通过点选左右两侧站点的名称以及上下左右的操作按钮来添加、删除站点以及调整站点顺序，点击保存后生成字符串传递给服务器端生成所需要的存储结构。IQC 录入以及保存检测站点数据的界面如图 10 所示，其余站点的界面类似。经测试，流程定制所需各项功能均已完成，运行稳定可靠。

Figure 9. User interface of process assembly  
图 9. 流程组装界面

Figure 10. User interface of test result data input and save for IQC  
图 10. IQC 检测结果数据的录入及保存界面

## 5. 结束语

本文聚焦于半导体 FT 测试 MES 中流程定制功能的类设计。首先尝试以传统的方法将功能需求抽象为三层树状结构，再将组合设计模式应用于类设计中从而大大提高了可扩展性，充分适应了半导体测试行业迭代速度较快的特点。在 Java Web 技术的基础上使用 Spring、JPA 框架对功能进行了实现，取得了良好的效果。

## 参考文献 (References)

- [1] 李贺明, 卫小松. 半导体企业 MES 的设计与实现[J]. 制造业自动化, 2015(24): 4-6, 22.
- [2] MESA International. MES Explained: A High Level Vision. MESA International White Paper Number 6, 1997.
- [3] 何月华. 半导体行业 MES 的研究与应用[D]: [硕士学位论文]. 上海: 上海交通大学, 2007.
- [4] 李杰, 彭月英. 抽象思维与设计模式[J]. 计算机与信息技术, 2011(6): 37-39.
- [5] 邵维忠, 梅宏. 统一建模语言 UML 述评[J]. 计算机研究与发展, 1999(4): 2-11.
- [6] 夏丽娟, 余青松. 基于组合设计模式的软件网络演化生长研究[J]. 计算机应用与软件, 2012(7): 133-136.
- [7] Riehle, D. (1997) Composite Design Patterns. *Proceedings of the 12th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 218-228. <https://doi.org/10.1145/263698.263739>
- [8] 祁耀斌, 李福太, 陈逢春, 等. J2EE 平台上 MVC 设计模式在开发电子政务系统中的应用[J]. 计算机应用研究, 2004(8): 203-205.

### 期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [csa@hanspub.org](mailto:csa@hanspub.org)