

**五、具体实施方式**

5.1 本发明完整的技术方案

本方案采用以下算法、流程来实现，主要分几大步骤进行：

1. 生成页面结构化文档

页面结构化文档是自动化测试技术的基础。在设计之初，需要根据既有的画面式样生成相应的页面结构化文档，该文档中定义了页面元素，页面元素的层次，页面元素的类型（显示类、点击类（有页面迁移、无页面迁移））及页面标志等信息，另外还储存了本UI画面中元素的基本信息、布局信息、动作数据。 每个页面对应一个页面结构化文档。

进一步的，由于画面式样的结构是固定的，因而可以通过分析页面结构化文档信息进行反向设计，显示相应页面，直观检验画面式样是否正确。

作为一个示例，页面结构化文档使用excel方法储存，文件系统结构按照App的不同进行区分。

1. 预设分类

预设脚本按共有预设、App专用预设分类。预设脚本之间设置关联预设索引，当关联预设索引存在的时候，同时将被关联的预设脚本一起生成预设脚本。在自动化生成测试用例的过程中，仅需在测试动作前拼接上预设脚本，就可以根据不同的预设，生成不同条件下的自动化测试脚本。

本实施例中，共有预设是对整个系统共通参数的设置，该类预设脚本可作用于任何测试脚本之前；而App专用预设是针对某一App专门设置的系统参数，因而该类预设脚本只能作用于特定App的测试脚本。预设脚本的分类可以减少不必要的测试脚本，降低测试成本。

1. 预设脚本生成

一种预设脚本的生成方法，包括以下步骤：

S1：根据业务需要，设计预设用例；

S2：通过分析页面结构化文档，将预设用例转化成预设脚本；

S3：对预设脚本按共有预设、App专用预设进行分类，并设置唯一索引；

S4：对于不同条件的预设用例，还需在预设脚本之间设置关联预设索引；

S5：将预设脚本与其相应的关联预设脚本进行拼接生成最终的预设脚本。

本实施例中，对于同一个页面元素的不同状态，每个状态都需要设计相对应的预设脚本；每条预设脚本都是对单一参数的设置，若想在一条预设脚本中完成对不同参数的修改，可通过设置关联索引的方式解决，同时会新增一条预设脚本。

1. 自动化测试脚本生成

如图所示，本发明提供一种自动化测试脚本的生成方法，包括：

S1：根据页面结构化文档中各页面元素的层次与类型，生成该页面相对应的元素树状图；

S2：根据页面结构化文档中页面ID与迁移ID，可以构建出整个系统的页面树状图；

S3：将S1得到的每个页面的元素树状图与S2得到的系统迁移的页面树状图合并，可以获得软件完整的树状图；

S4：利用深度优先算法遍历S3获得的树状图，从根节点到叶子节点的每一条路径都将生成与之对应的一般测试脚本；

S5：依次将既有的预设脚本拼接到一般测试脚本之前，以生成最后的测试脚本；

在本实施例一种可能的实现方式中，步骤S4中的根节点可以为软件Home画面，叶子节点既可以是元素类型为显示类的页面元素，也可以是当迁移ID存在于当前路径上的点击类元素。

本实施例中，自动化测试脚本的生成参照的是页面结构化文档，而非软件运行时页面结构，因而元素树状图中包含了当前条件下画面上未能表示出页面元素，从而保证了测试脚本的完整性。

1. 自动化测试实施
2. 自动化测试结果比对
3. 自动化测试结果生成
4. 功能覆盖率计算

5.2 本发明技术方案带来的有益效果

