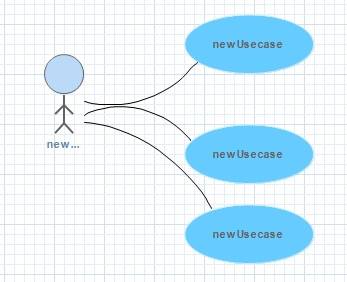
1. **概述**

客户端能够实现用例图、框架型视图、流程型视图等图表的绘制工作，能通过与服务器交互，来完成版本上传、版本回退、版本差异比较、冲突检测与解决等功能。

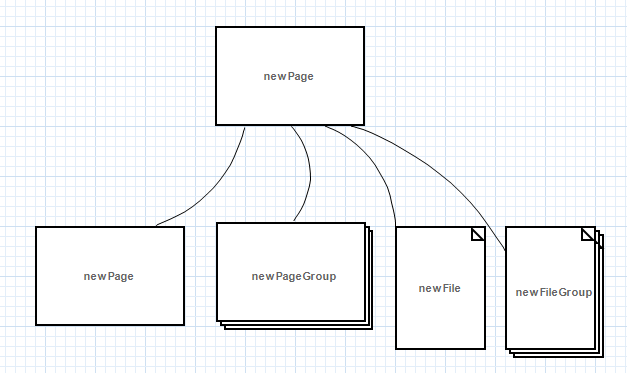
1. **功能描述**

**2.1 图表绘制**

1. 客户端启动后能选择新建项目，在新建的项目中能选择绘制何种图形，不同的图形所能绘制的图形元素和图形结构不同。
   1. 用例图



* 1. 框架型视图



* 1. 流程型视图



1. 完成图表的绘制后，客户端可以进行保存操作，将所绘制的图表持久化为文本文件。

**2.2 图表间的跳转**

1. 客户端在绘制图表时，可以将一个图表中的图形元素与另一张图表进行关联；
2. 完成了关联操作后，可以从该图表跳转导航至与之关联的图表，反之亦可。

**2.3 版本上传**

1. 客户端在保存好图表后，可以对整个工程目录进行上传操作；
2. 点击上传的菜单按钮后，客户端将工程内的所有图表转换成为中间文件；
3. 中间文件转换完成后，将所有的中间文件，分批次的通过HTTP协议，发送给服务器进行处理；
4. 服务器收到所有的中间文件后，调用冲突检测功能进行冲突检测（参见2.6），如没有发生冲突，则将客户端上传的文件储存至数据库中，并向客户端返回成功的操作结果。如发生了冲突，则返回冲突的操作结果，并将冲突的描述发送给客户端。

**2.4 版本回退**

1. 客户端在点击了版本回退的按钮时，向服务器发送版本回退的请求；
2. 服务器收到版本回退的请求后，向数据库中查询当前已有的版本，返回版本列表给客户端；
3. 客户端收到版本列表后，展示给用户，用户选择回退至某一版本号的版本；客户端将该版本号发送至服务器；
4. 服务器将该版本号对应的中间文件发送给客户端；
5. 客户端将中间文件转换为图表文件，覆盖当前的工程。

**2.5 版本差异比较**

1. 用户点击版本差异比较按钮，客户端提示用户选择两个进行比较的版本；
2. 用户选择好用于比较的两个版本后，客户端将差异比较的请求发送至服务器；
3. 服务器执行版本差异比较功能，返回比较结果；
4. 客户端将比较结果以图形化形式展示给用户。
   1. **冲突检测及解决**

准则一：冲突只存在于同一版本基础上的两个不同修改方案之间。

准则二：满足以下两种情况之一则认定为冲突：

1. 两个方案对同一ID的元素的同一个属性进行了不同修改，且该属性不是一个数组。
2. 一个方案删除了某个元素，而另一方案对同一个元素的属性进行了修改

准则三：图形元素的坐标、大小不计入冲突范围。

可能发生修改的元素包括：图表，节点，连线。

可能发生冲突的元素只有节点。

对于某个图表，可改动的属性包括：

1. 节点列表nodeList
2. 连线列表connectionList

对于某个节点，可改动的属性包括：

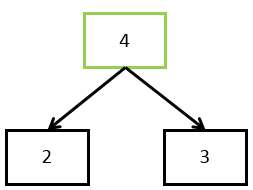
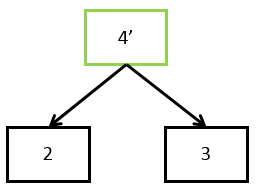
1. 出连线outgoing
2. 入连线incomming
3. 文字text
4. 子节点列表nodeList
5. 跳转关系冲突

对于某个连线，可改动的属性包括：

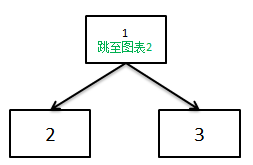
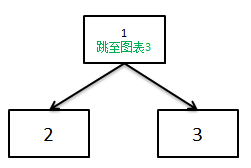
1. [起始,终止] 节点id对[start, end]

// 注：由于该属性唯一标识了一条连线，因此该属性的修改等效于删除原有连线并创建一条新的连线

* 冲突举例
  + 某节点的文字被同时改动

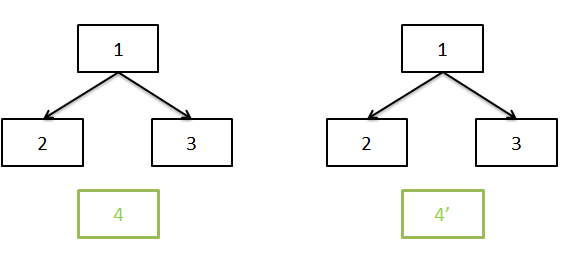
与发生冲突，冲突点在原节点1的文字，两个版本都修改了该节点的文字；

* + 某节点的跳转关系被同时改动

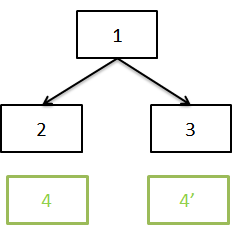
与发生了冲突，冲突点在节点1的跳转关系上；

* 不冲突举例
  + 某图表的属性改动

图表的节点列表被分别添加了4和4’：

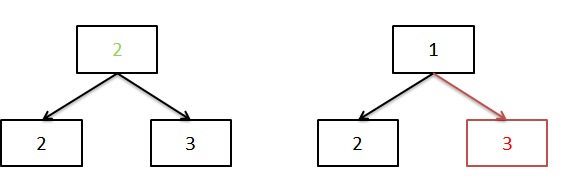


合并结果：

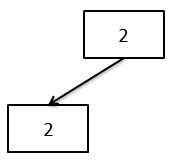


* + 某节点的不同属性被分别改动

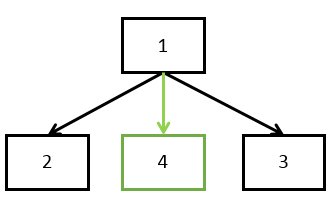
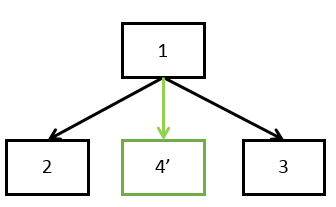
节点1的文字和出连线被分别修改：

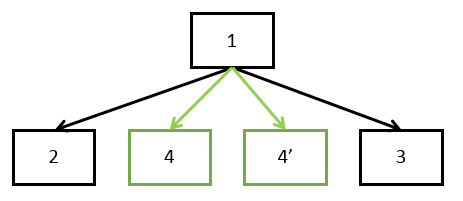


合并结果：

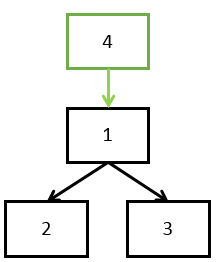
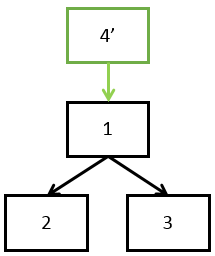


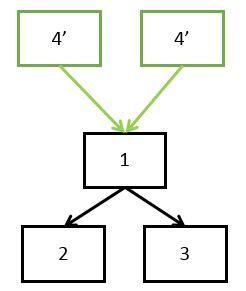
* + 某节点的出连线被同时改动

与不冲突，合并后结果：

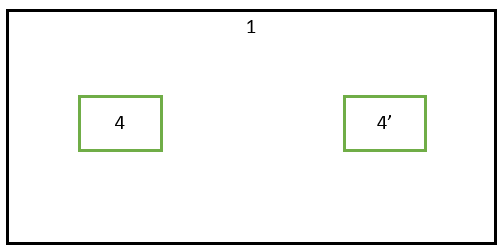


* + 某节点的入连线被同时改动

与不冲突，合并后结果为：



* + 某节点的子节点列表被同时改动



不冲突，可以同时保留这两个新加入的节点。

1. 当服务器收到客户端的提交请求时，根据以上的冲突检测逻辑，来判断客户端本次上传的版本是否与历史版本发生冲突，并尝试进行合并操作；
2. 如发生了冲突，则服务器将发生冲突的信息（冲突点、冲突描述、冲突列表、合并信息）以冲突描述文件的形式发送给客户端，客户端解析该冲突描述文件，将冲突以可视化的形式展示给用户，并引导用户通过点选的形式来解决冲突；客户端将操作的结果反馈至服务器端；
3. 如不发生冲突，服务器则执行持久化操作、推进版本号等，向客户端返回操作成功的结果。