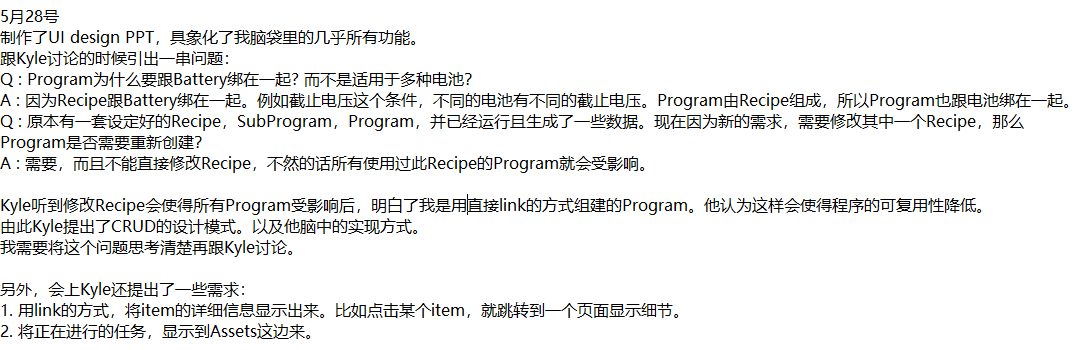
原来的设计，Program创建时是跟Battery Type绑在一起的，所以当时很自然的提出了Estimated Time这个概念。

~~现在的设计，Program用哪一个Battery，是在Execute的时候决定的，也就是说同一个Program （sub program），可能使用不同的电池。不同的电池肯定有不同的时间，必须分开记录~~。

~~预估时间是在Execute时获得，因为Execute时才知道用的什么电池~~。------在Program创建时，确定Battery Type，如此一来，创建时便知道时间。

*回顾一下当初将Battery Type剥离的原因：*

**

*那个时候绑得很紧是因为Program跟SubProgram是Link起来的，而SubProgram里的Recipe的TestSteps又跟Battery Type强相关，所以整个绑住了。现在一来SubProgram是从Template创建的，二来TestSteps跟Battery Type无关。所以Program的Battery Type是可以改变的（可以在Save As操作时随意改变）。*

*因此，我们需要在Program里加上Battery Type，从而在Program创建时确定预估时间。*

预估时间在Commit时更新。

现有设计还有一个地方需要调整，那就是sub program template需要加上loop属性用来描述循环次数。不然的话，无法区分循环次数不一样的sub。如果循环次数不一样，预估时间当然也不一样。不要预估时间的时候可以不管，要预估时间的时候，就得区分了。具体计算方式见方案一和方案二

~~又因为Execute和Commit是对subprogram执行的操作，而我们需要将预估时间更新到sub template上，所以subprogram需要将sub template作为自己的一个私有属性。~~

以下方案，是针对历史数据而言：

方案一：

建立一个表格：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sub Program Template Id | Test Count | Battery Type Id | Executed Count | Average Time |
| 1 | 1 | 1 | 5 | 3.5 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

每做完一个test（commit），就更新这个表。具体方式是填入前三个ID作为索引，

AverageTime = (AverageTime \* ExcutedCount + CurrentTime/loop)/( ExecutedCount+1) ExecutedCount+=1;

每次Execute时，根据这三个索引，查询出AverageTime\*loop.

方案二：

可不可以直接从TestRecord获得预估时间？

假设TestRecord记录了自己创生于哪个subprogramtemplate的哪个testcount。

在dbContext.TestRecords中找出所有对应subprogramtemplate, batterytype, testcount的记录。这样可以获得executed count（sum(o=>o.loop)）和总时间。这样就可以获得平均时间，也就是预估时间。再结合当前loop算出具体预估时间为loop倍

这样做的好处是不需要改动太多现有代码。风险是当数据量非常大的时候，查询需要的时间也随之增多。

考虑长期维护方便，采用方案一。

以下为预估时间初始化：

可是如果没有历史数据可供参考呢？那就要用电池规格和实验条件来判断。

T = C/I\*coff

例如一个3000mAH的电池，以200mA放电，则放电时间为3000/200=15小时。因为充电电流会变小，所以时间会比15小时长。为了简便，取一个系数1.2，即预估时间为15小时\*1.2=18小时。

放电的情况也类似，也是容量/电流\*系数。系数也取1.2

C的设置，是在电池创建时完成的。

C的获取，是在Create或Save As Program的时候，给定Battery Type之后确定的。

I的获取相对来说比较麻烦，要分几种情况。

1. 如果是充电电流，那么一定是Capacity Rate。值的范围在0~1之间。cr\*c=I
2. 如果是放电电流
   1. 如果类型为Capacity Rate。则与充电电流相同
   2. 如果类型为mA，则直接使用这个值
   3. 如果类型为Dynamic，则没有值（因为Dynamic的描述很复杂）

因此，我们要将Subprogram的Discharge Current，拆分成DischargeUnit和DischargeCurrent两项

|  |  |
| --- | --- |
| DischargeUnit | DischargeCurrent |
| % | 0~1 |
| mA | N |
| Dynamic | NA |