按V5 change list和V5 design doc设计好model之后， 开始用MVVM来设计viewmodel和view。

不过首先，原来的model设计中少了repository。使用repository的好处是，可以将序列化/反序列化封装起来。也为集合view提供了model。

当用户在UI上点击命令想要创建model时，我们首先新建一个空的model，然后关联一个新的viewmodel，并作为workspace添加到工作区。于是用户在UI的workspace上看到新的viewmodel。用户改变UI项目时，viewmodel属性随之改变，同时model的数据也相应改变。点击save后，判断model合法性，并将其加入repository。

因为allviewmodel订阅了repository的ItemAdded事件，所以将model加入repository的同时，我们也在allviewmodel中加入viewmodel。

设计Scheduler的时候，有点犹豫要不要把scheduler看做纯粹的viewmodel。因为它是requestRepository的某种展示，是requestRepository的consumer，很像AllRequestsViewModel。但是转念一想，一个task要放到哪个list里（waiting, ready, running, complete, abandoned）,是属于model层面的逻辑。所以暂时还是认为scheduler是model。

Scheduler的viewmodel设计：

Recipe的executor有自己的viewmodel。几个按钮和几个combobox，editbox。

主结构有两种方式：

1. treeview，对应几个list。Treeview里面展示subprogram和recipe的关系。
2. list，放所有的executor。

方案2比较简单，所以先尝试看看。

具体步骤为：

1. Request被添加到requestRepository的时候，意味着它生效了，所以它附带的executor要被添加到executorRepository。所以Repositories要订阅requestRepository的ItemAdded事件。
2. AllExecutorViewModel初始化的时候就要显示已有的executors，所以要用CreateAllExecutors来将executorRepository中的数据导入到viewmodel中来。

原本的scheduler，其实既有model的逻辑，又有view的东西。

更好的方式应该是：

1. Executor的状态切换逻辑，Executor之间的关联性，这些属于model
2. 对Executor的展示，不管是DataGrid还是TreeView，都属于ViewModel和View

因为Executor是Request的成员属性，所以可以不用executorRepository. 不用Scheduler。只要有requestRepository就够了。不管是展示executor的datagrid还是treeview，都可以从requestRepository构建其viewmodel