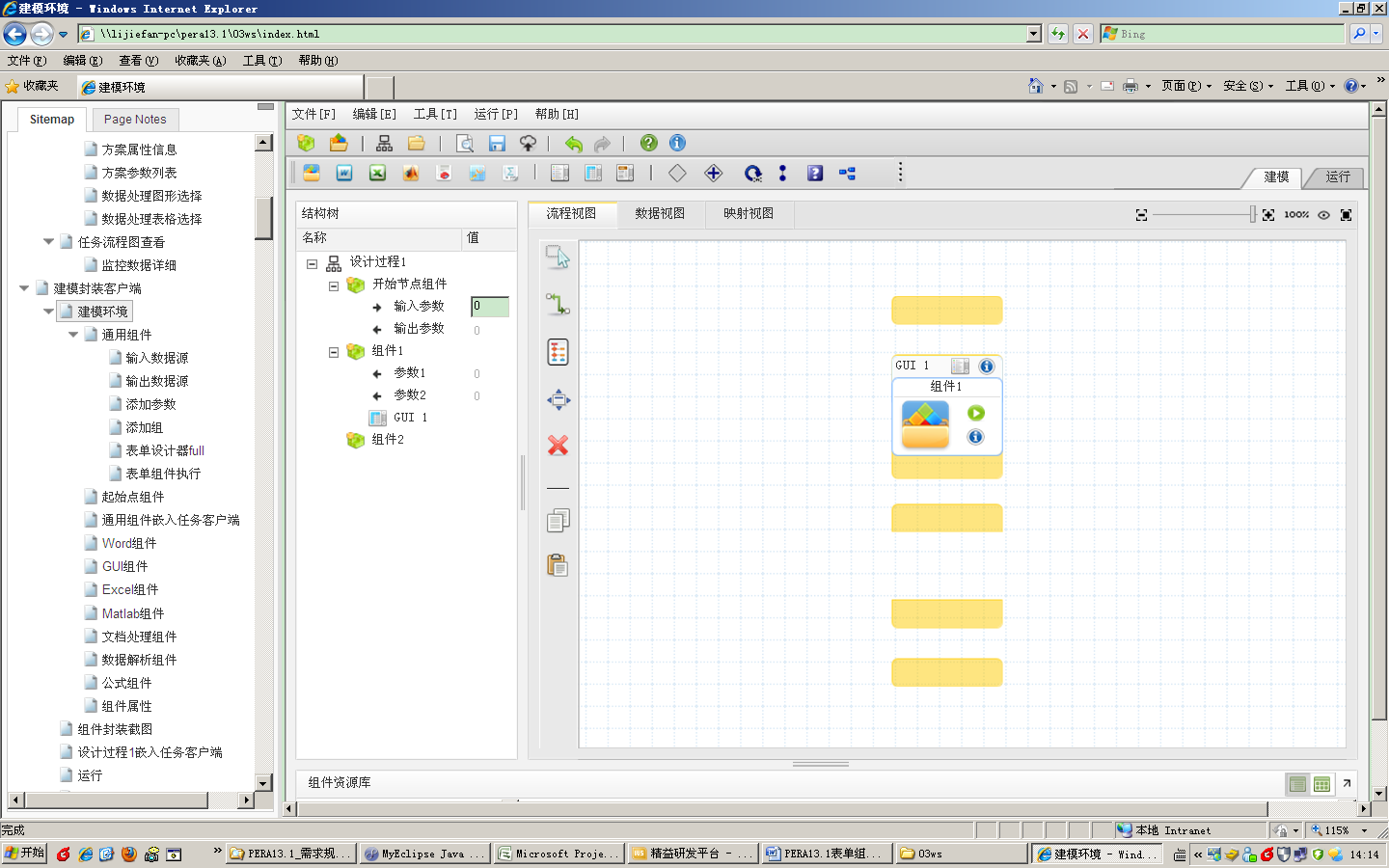
表单组件在综合设计WS的设计与应用

2013-6-9 杨波

1. 需求

对于表单组件，要求可以在开始节点上设置，并且对应全体组件的输入和输出。同时单个节点可以设置前后表单，对应该节点的参数展示。执行时表单必须在用户本地弹出。下图为建模时拖拽表单组件的效果，即节点会显示前后可放置的高亮区域。

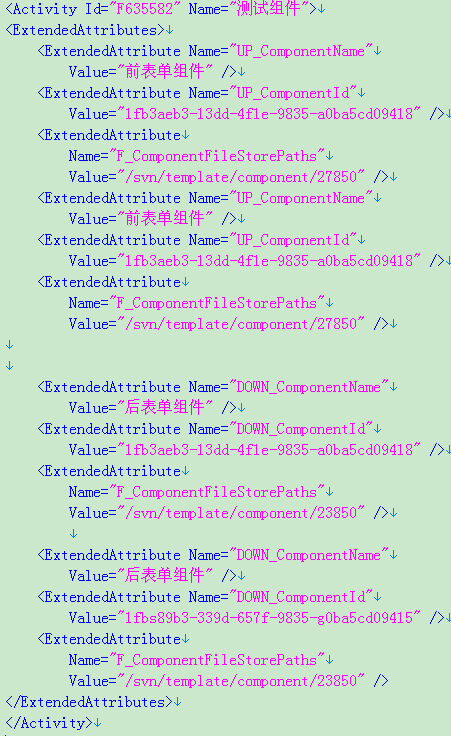


1. 定义

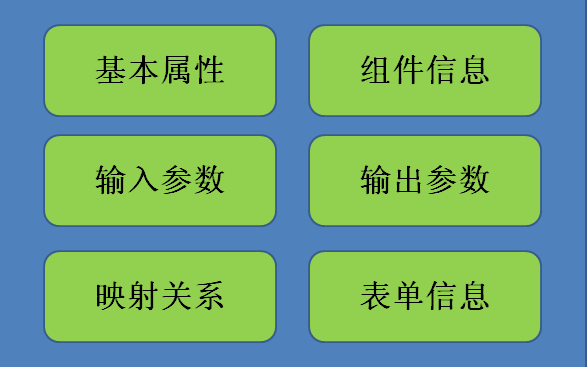
* 表单组件在建模中，属于任务节点的属性，不能独立使用。
* 表单组件的参数等同于组件的参数。表单组件不对外公开，即数据映射时看到的仍为组件的参数，同时表单组件也不在流程结构树上展示（此处需要再次讨论）。仅将表单组件视为组件数据录入的UI。
* 提供开始节点特殊的设置功能：前后表单定义，用于初始全体参数的录入及执行完成后全体参数值的展示。
* 表单组件在封装环境编辑时，不能对参数进行编辑操作。
* 后表单执行时展示的数据只能查看，不能修改。
* 前表单的数据展示后可以修改，因此执行时需要如果有数据映射，则应该以表单实际输入的值作为执行值（此处需要再次讨论）。

1. 前后表单的数据结构

对于活动的XPDL定义，增加扩展属性，记录表单相关信息。对于表单参数和组件参数的关系，同样记录在此扩展属性中。具体与客户端协定。强制要求表单组件的参数ID和节点保持一致。表单属性中记录所属节点的全部参数ID。以下为XPDL定义格式：



以下为组件节点属性结构图：

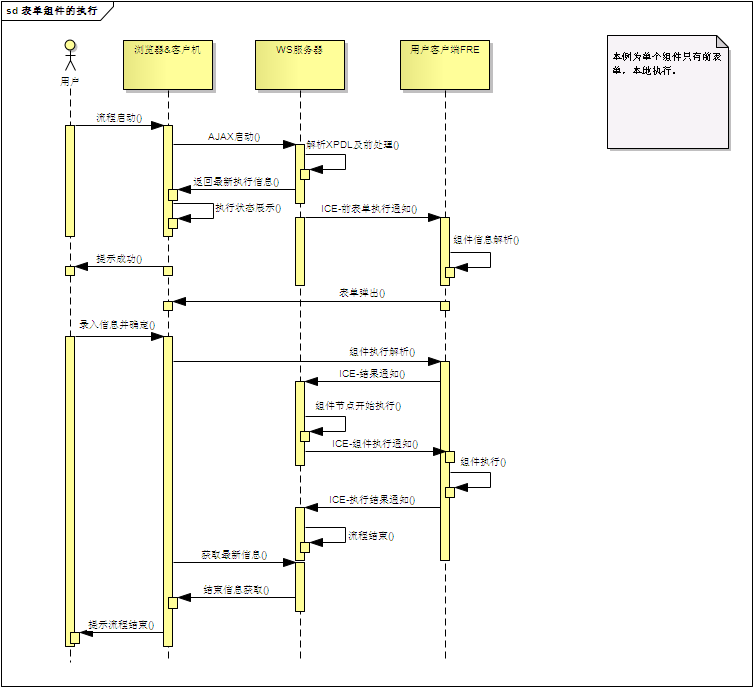


1. 前后表单的建模

对于表单组件，参考最新UI。有表单的节点在图形展示上与其他节点区分开。实际设计效果等待UI确定后设计。建议采用右键菜单增加和编辑的功能，简化操作。

1. 前后表单的引擎处理

对于表单组件，在PeraXpdlNode增加属性记录（参考上述的XPDL片段）。可考虑对于活动，增加前处理和后处理。在此类接口中完成表单的相关操作。后续变更仅对此接口进行相应的修正而不影响驱动逻辑。处理时序图如下：



对于前处理和后处理的设计如下：

* 前处理：

前处理在createTask方法进行。主要是负责解析当前节点是否具有前表单。如果有，则将此表单组件信息放入特定的全局变量。由于存在并行节点，因此变量应为集合。待引擎驱动完成后，再将此集合信息统一以ICE通信的方式通知指定的客户端FRE环境。同时，节点的状态应该仍然为执行中，但是开始时间字段为空，即等待表单输入不统计为组件执行时间。待表单输入完成后，服务器接到FRE通知的表单结果信息，通过节点唯一标识，开始实际执行节点。

* 后处理：

后处理在finishTask方法进行。主要是负责解析当前节点是否具有后表单。如果有，则将此表单组件信息放入特定的全局变量。此变量即为上述前处理变量。即前后表单均同时通知FRE。此时当前节点为已完成状态，执行时间也统计完成。当弹出后表单时，用户对于此表单页面的确认或关闭操作不影响流程执行。即后表单仅为结果数据的通知和提示，因此会存在用户环境同时弹出前后表单的情况（此处需要再次讨论）。

1. 执行环境接口

对于表单组件的执行，需要客户端提供特定的接口。同样的，WS服务器也提供特定的表单结果提交的接口。接口参数除去普通节点同样的相关参数外，还需要任务ID。此处可参考组件执行和提交时的ICE通信接口的参数格式。

1. 表单和组件匹配的校验

* 对于节点修正参数后，如果已定义表单组件，考虑采用对比的方式，确认表单是否正确。可以将表单参数全部以扩展属性记录在XPDL中，包括名称，ID，类型等。但是此方法会导致XPDL文件存在参数冗余数据。需要讨论是否有更好的方法。
* 表单和组件的参数不匹配的，在节点上以图标形式提醒。

1. 开始节点上的前后表单

开始节点设置前后表单，理解为在执行初始和结束，系统提供一个全部参数录入和展示的功能。其数据结构可以参考节点的表单格式，区别仅在于挂接在基础流程定义下。执行时，同样在引擎发起和结束时增加额外处理。但是开始节点的表单同样存在匹配校验的问题，如果采用全部参数记录对比的话，那么系统冗余数据非常多，理论上会导致整个XPDL文件异常庞大，影响性能和效率（缓存在服务器的XPDL对象容量增大）。例如，过程中有10个节点，每个节点有10个输入和10输出，每个节点设置前后表单，同时设置开始节点的前后表单，则流程定义中参数要记录600个，其中包括冗余数据400（开始节点记录的100个输入和100个输出，还有每个节点记录的10个输入和10个输出）。