# 中间标签：

对于一句话(允许关键词)，标签设计如下：

* 内容
  + Given
    - Precondition
      * PreconditionType
    - fixed data
  + When
    - action
      * position
      * actionType
    - input data
  + Then
    - Postcondition
      * postconditionType
    - output data

对于一个GWT，标签设计如下：

* Use Case:
* Name
* BranchScenarios
  + BranchScenario
    - Name
    - BranchConditions
      * Branchcondition
* preScenarios:
  + preScenario:
    - Name
    - conditions
      * condition
* postScenarios
  + postScenario:
    - Name
    - conditions
      * condition
* FlowType: Basic Flow/Specific Alternative Flow/Global Alternative Flow/Bounded Alternative Flow

# 中间标签描述：

对于一句话：

Scenario: 所在Scenario标识

其他的均为在GWT文档中的唯一标识记录

# GWT->中间标签规则：

1. 输入文档存储时，将输入的GWT按照feature,scenario,given,when,then四个部分进行存储。
2. 将GWT中原先的每一行作为一个句子来进行拆分
3. 给该句话指定初始标签，包括对于语句的类型进行标注，对于INCLUDE、 EXTENDED、GLOBAL、 Use Case Name这类具有关键词的语句进行标注
4. 如果该初始标签处于需要自然语言处理的标签列表中（例如该句话是Given中的一句肯定句或否定句（在这里，因为我们设计GWT中无分支，所以action在这里先不用进行处理），则进行以下过程，否则直接进行接下来的过程：
   1. 对语义相近句子进行统一化，对该语句进行独一相近标注，并修改其Type值
   2. 对语义相反语句进行规范化，对这两个语句进行独一相反标注，并修改其Type值
5. 根据GWT中的单个语句与其从属Use Case，按照一下规则对GWT进行标签化：
   1. 给定GWT文档G1，G2
   2. 提取两个文档中，标签中存在precondition和postcondition的语句
   3. 判断条件
      1. 若G1与G2中，标签为precondition的语句，部分语义相同，部分语义相反，则G1与G2彼此添加为BrachScenario标签，并且添加用于进行分支的conditions
      2. 若G1中标签为precondition的语句与G2中标签为postcondition的语句完全完全相同，则G1的preScenario为G2，G2的postScenario为G1

# 中间标签->RUCM规则：

1. 下面对Use Case标签相同的GWT进行处理（以下描述均为正常状况）
2. 提取Use Case标签作为RUCM的Use case Name
3. 将GWT的scenario描述进行自然语言处理，进行结合，形成Brief Description
4. 提取无preScenario的GWT的所有标签为precondition的语句作为precondition
5. 提取无preScenario的GWT的所有标签为action的语句中的人物提取，作为Actor，存储以便于标准化
6. 提取GWT中的关键词，放至dependency和Gerneralization
7. 将无preScenario的GWT的所有标签为action的语句，按照action中的position标签放到Basic Flow的最前面。
8. 以无preScenario的GWT的postScenario作为线索，形成整个BasicFlow，与此同时进行Actor名称的统一化表述
9. Basic Flow的postcondition为最后一个添加的GWT的所有标签为postcondition的语句

10、以Basic Flow中的GWT作为线索，寻找其BranchScenario标签，从而填写其他三种类型的Flow，并且修改Basic Flow原始语句，添加Validate that 等关键词。