

使用Drupal Workflow模块做提案管理

上海外国语大学 张文正

wzzhang@shisu.edu.cn

2015-10-21

需求

基本流程：

1. 提案员提交提案->
2. 提案工作组审核提案（立案/退回修改/不予立案）->
3. 提案工作组分配主管校长和职能部门回复提案->
4. 提案员对处理结果的回复和评价

用到的模块

- Workflow及其子模块
- Workflow fields
- Rules
- Fivestar/Entity API

操作1： 安装启用模块（2.x版）

- 必选：
 - Workflow api
 - Workflow field
 - Workflow ui
 - Workflow views
 - Workflow fields
- 可选：
 - Workflow rules
 - Workflow actions

操作2： 添加提案内容类型

可能有的字段：

- 标题
- 提案发起人
- 附议人
- 提案类别
- 届次
- 提案内容
- 建议解决措施
- 主管校领导
- 主管校领导批示
- 负责职能部门
- 职能部门答复意见
- 提案发起人对各方处理的意见
- 提案发起人对各方处理的评价
- 提案状态

workflow field

操作3：添加 workflow

1. 添加 workflow
2. 添加 workflow 状态和用户角色
3. 设置 workflow 转换矩阵
4. 在 node 中添加 workflow field，关联该 workflow（即提案状态字段）
5. 设置每种状态下的 node 中每个字段对作者和各种角色的可见性和可编辑性

操作4： 添加提案管理视图

- 我发起的提案
- 我要处理的提案（主管校领导和职能部门领导）
- 提案工作组可管理的提案（`workflow_vbo`）

workflow vs workbench?

Workflow提供了三样workbench不具备的功能:

1. 状态切换矩阵
2. Workflow fields字段级别的可见性和可编辑性设置
3. Workflow本身是一个field, 可以加入任何entity

在没有上述需求的情况下, Workbench及其扩展模块是更佳的内容生命周期管理模块, 尤其是:

1. 涉及到按部门和按用户角色的条块分割式权限控制时更合适。(workbench access)
2. 基于[状态机](#)理论的workbench moderation 2.x版, 更加灵活

扩展todo

- Workflow + Rules 配合
- 对同一内容类型添加多个 workflow
- Workflow access 内容权限管理