Acumatica Workflow Engine

Notation and Usage Examples

Stan Lesin
Senior Developer
Acumatica
Agenda

- Workflow Fundamentals
- Our Automation Engine Issues
- The New Acumatica Workflow Engine
- A Workflow Customization Example
- Demo
- Resources
- Summary
What is Workflow?

Visual representation of the document life cycle
Acumatica Automation Engine Issues

Not an obvious mechanism and structure

Not possible to make changes from code

Automation customization is not recommended

Hard to upgrade in case of local changes

Obscure inter-screen interactions
What is Our Workflow Engine?

It is a replacement for the Automation Engine.
Workflow Mechanism – Plain Old Finite State Machine

**Shipments**

**SO302000 Workflow**

- **On Hold**
  - Init: SOSetup.holdShipments = true
  - Release from Hold
  - Put on Hold
- **Open**
  - Init: SOSetup.holdShipments = false
  - Confirm
- **Confirmed**
- **Partially Invoiced**
  - On Invoice Unlinked
  - On Invoice Linked
- **Invoiced**
  - On Invoice Unlinked
  - On Invoice Linked
  - Correct
  - On Invoice Released
- **Completed**
  - On Invoice Cancelled
Workflow Structure

Screens

- Default Field States (IsHidden, IsDisabled,IsRequired, DefaultValue, ComboBoxValues)
- Action Definitions (Placement, Mass Process Exposing, Field Assignments)
- Event Handler Definitions (Event Binding, Entity Provider, Field Assignments)
- Condition Definitions
- Workflows (based on Document Type)
  - Flow States (based on Document Status)
    - Actions (which are present in the current flow state)
    - Field States (which alter default field states in the current flow state)
- Flow Transitions
  - Origin and Target Flow States
  - Trigger with optional Condition (Action or Event Handler)
  - Field Assignments
Workflow is defined in code

class Workflow
{
    public void Define()
    {
        context.AddScreenConfigurationFor(screen =>
            screen
            .StateIdentifierIs(status =>)
            .AddDefaultFlow(flow =>
                flow
                .WithFlowStates(fss =>
                    {
                        fss.Add(initialState, flowState => flowState.IsInitial(g => g.initializeState));
                        fss.Add(state.hold, flowState =>...);
                        fss.Add(state.open, flowState =>...);
                        fss.Add(state.confirmed, flowState =>...);
                        fss.Add(state.partiallyInvoiced, flowState =>...);
                        fss.Add(state.invoiced, flowState =>...);
                        fss.Add(state.completed, flowState =>...);
                    }
                .WithTransitions(transitions =>
                    {
                        transitions.AddGroupFrom(initialState, ts =>...);
                        transitions.AddGroupFrom(state.hold, ts =>...);
                        transitions.AddGroupFrom(state.open, ts =>
                            {
                                ts.Add(t => t.ToState.hold().IsTriggeredOn(g => g.putOnHold).WithFieldAssignments(fas => fas.AddHold(f => f.SetFromValue(true))));
                                ts.Add(t => t.ToState.confirmed().IsTriggeredOn(g => g.confirmShipmentAction).When(conditions.IsConfirmed));
                            }
                        );
                        transitions.AddGroupFrom(state.confirmed, ts =>
                            {
                                ts.Add(t => t.ToState.open().IsTriggeredOn(g => g.confirmShipmentAction));
                                ts.Add(t => t.ToState.invoiced().IsTriggeredOn(g => g.OnInvoiceLinkedHandler).When(conditions.IsInvoiced));
                                ts.Add(t => t.ToState.partiallyInvoiced().IsTriggeredOn(g => g.OnInvoiceLinkedHandler).When(conditions.IsPartiallyInvoiced));
                            }
                        );
                        transitions.AddGroupFrom(state.partiallyInvoiced, ts =>...);
                        transitions.AddGroupFrom(state.invoiced, ts =>...);
                        transitions.AddGroupFrom(state.completed, ts =>...);
                    }
                );
            });
        }
    }
}
Workflow Customization Example – Shipment On Inspection State

Shipment SO302000
Workflow + Inspection State

Initial (non-persistent) -> On Hold

- Init*
- Put on Hold

On Hold -> On Inspection

- Release from Hold
- Inspection Failed
  - Put On Inspection
  - Inspection Passed

On Inspection

Open

- Confirmed
- Correct

SOSetup.holdShipments = true

SOSetup.holdShipments = false
Demo — Workflow Customization Example
Summary

Please add some bullets summarizing your presentation...
Available Resources

Please add links to resources that devs will find useful here.