

JCore - Technical Background

Task:

The following document shall explain the major structure of the JCore and its functionality.

Realization:

The *ExecutionService* is the class that handles the complete *JCore*. The *ExecutionService* manages all *ScenarioInstances* and provides methods to interact with them, for example executing an activity. The class *ScenarioInstance* represents an instance of a PCM Scenario. It saves all its *FragmentInstances*, *ControlNodeInstances* and *DataObjectInstances*. If a *ScenarioInstance* gets initialized, it initializes all its *FragmentInstances* and *DataObjectInstances*. *DataObjectInstance* represents an instance of a PCM Data Object. It has a state and a list of all its *DataAttributeInstances*, instances of Data Attributes, which save the value and the type of the attribute.

The *FragmentInstances* are responsible for initializing all *ControlNodeInstances* belonging to the corresponding Fragment. *ControlNodeInstances* might appear as *ActivityInstances* or *GatewayInstances*, as shown in the PCM model. All *ControlNodeInstances* have a *StateMachine*, an *IncomingBehavior* and an *OutgoingBehavior*. The *StateMachine* controls the state of the *ControlNodeInstance* and handles the state transitions. The *IncomingBehavior* controls the incoming control flow. This is important for Gateways, because this behavior decides if a Parallel Gateway can be enabled. The *OutgoingBehavior* controls the outgoing control flow, including the Data Object output states for an activity, which are set here.

An *ActivityInstance* also has a *TaskExecutionBehavior* managing further consequences of executing the activity. This may be the *HumanTaskExecutionBehavior* that sets the Data Attribute states or a Service Task execution behavior like the *EmailTaskExecutionBehavior* that automatically sends an email.

Visualization:

