Behaviors and Events

This chapter explains how the Event-System in Behaviors work.

General

Behaviors communicate via Events with each other and with other components of the application. For each action a specific event is implemented. An Event can carry any data or none at all. For a custom Event simply derive from FeatStd::Event. Each Behavior can return an event-result after handling an Event in the "OnEvent"-method. Behaviors are using unique identifiers (FeatStd::Internal::Identifier) to identify the sender of an event.

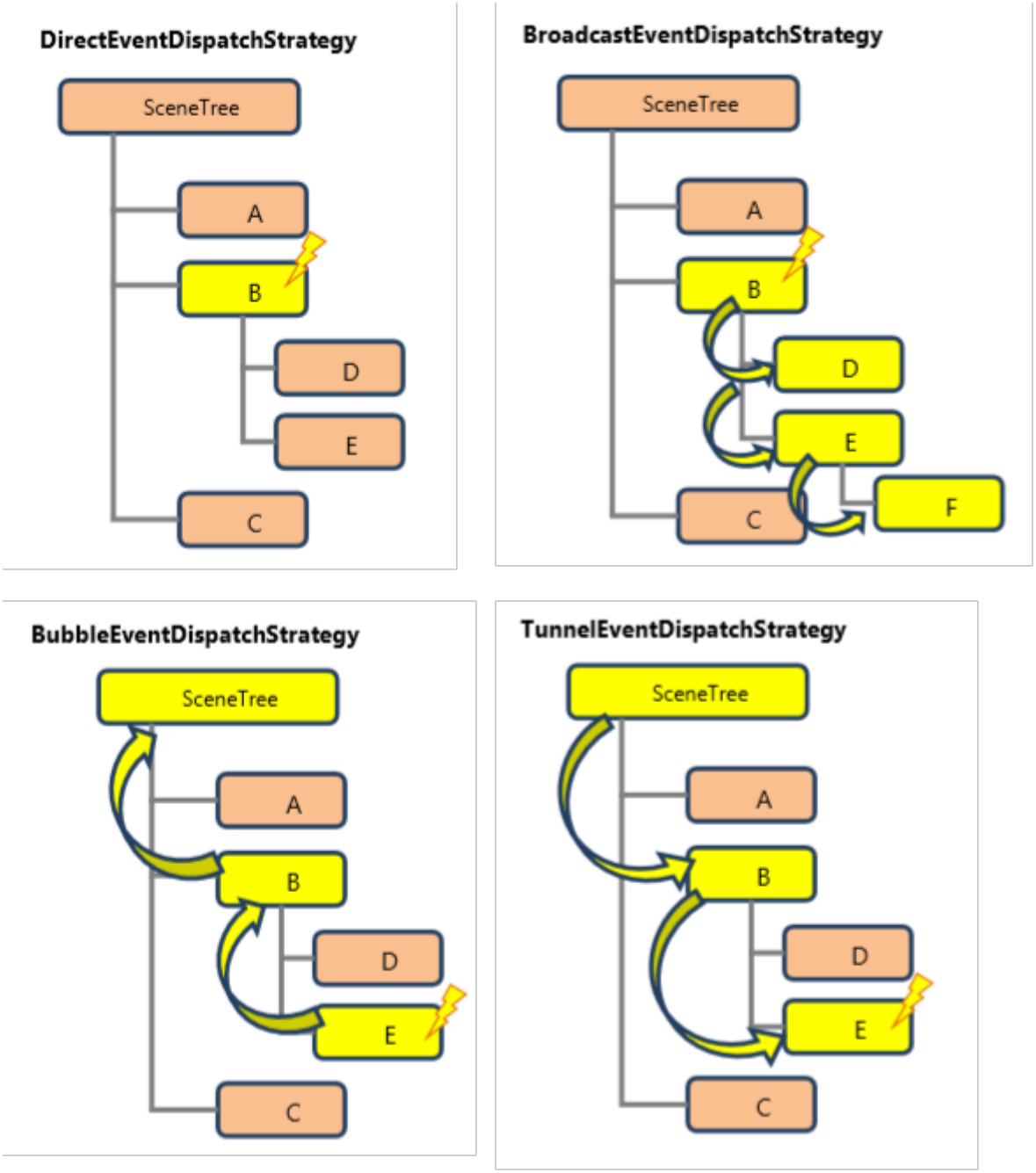
```
        void OnPressedEvent(const PressedEvent& event, Candera::EventDispatchResult& /*dispatchR
        {
            if (event.GetSender() == m_behavior.GetIdentifier()) {
                static_cast<void>(m_behavior.SetState(ControlStateEnum::Pressed
, event.GetType() == PressedEventType::Pressed));
            }
        }
```

For a list with all Events which are used by Behaviors also check chapter Control Behaviors Events . There you can also find their readable names (names in SC).

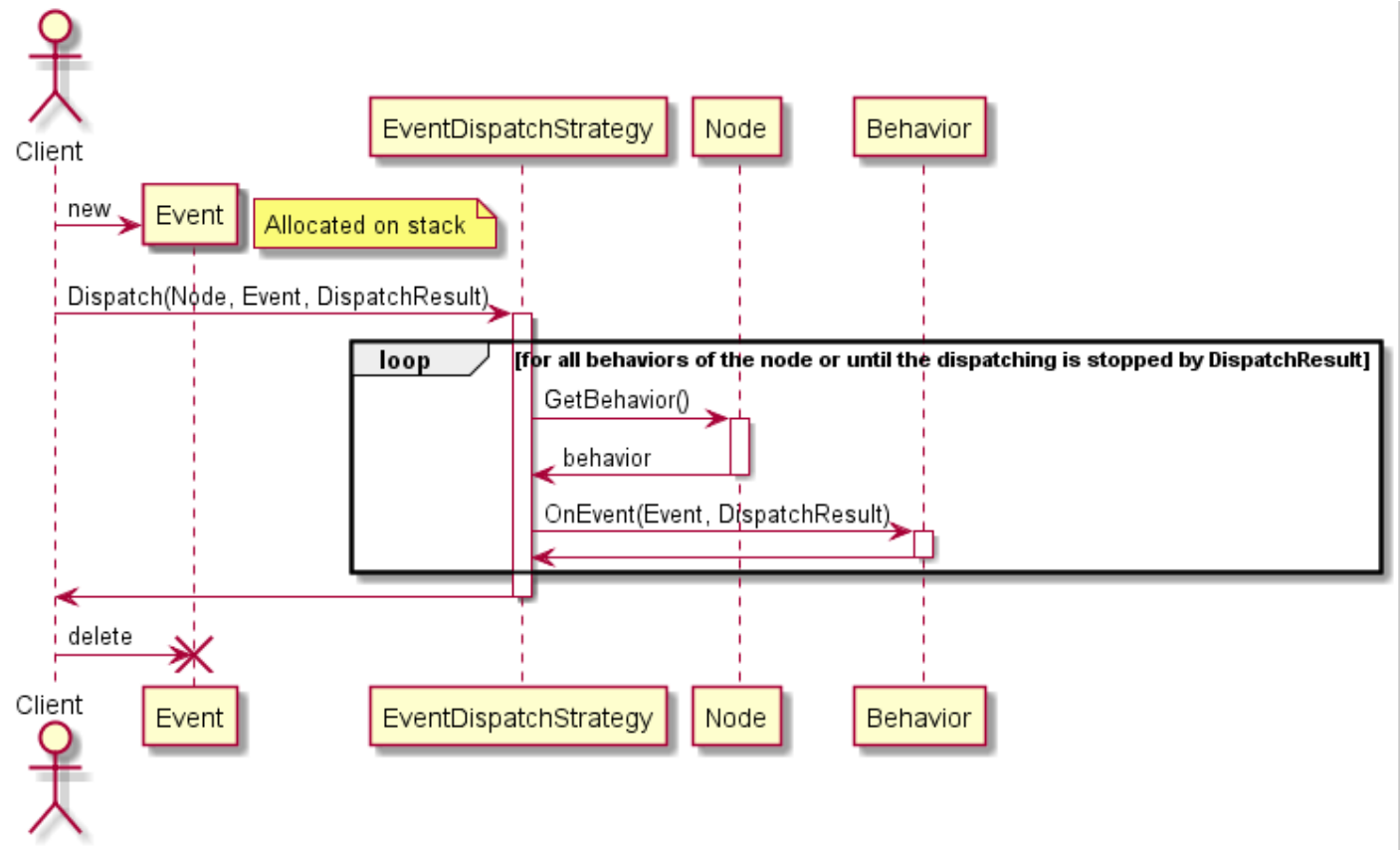
Event-dispatching

The Behavior base-class implements *DispatchEvent*-functions. Events are dispatched on a Node using one of the available *EventDispatchStrategy* objects. An Event is passed to all OnEvent function of all Behaviors. The following dispatch strategies are available:

- **DirectEventDispatchStrategy:** Only the behaviors of the node will receive the event.
- **BroadcastEventDispatchStrategy:** The event will be first dispatched to the current node then to all child nodes.
- **BubbleEventDispatchStrategy:** The event will be dispatched from the current node to the parent and so on until the root node is reached.
- **TunnelEventDispatchStrategy:** The event will be dispatched from the root node to the current node.



Event Dispatching Diagram



Control Behaviors Events

This chapter gives an overview of the most important control behaviors events. Furthermore they are listed with their readable names (names in SC). For further information of how to use them in SC see chapter EventHandler Tab .

Events

Class Candera::ProcessValueEvent
ProcessValueEvent - Value Processed

Class CgiStudioControl::AnimationEvent
AnimationEvent - Animation State Changed

Class CgiStudioControl::ArTargetEvent
ArTargetEvent - ArTargetEvent event

Class CgiStudioControl::CaretPositionChangedEvent

CaretPositionChangedEvent - Caret Position Changed

Class CgiStudioControl::ChangePropertyEvent

ChangePropertyEvent - Property Changed

Class CgiStudioControl::ChangeSelectionEvent

ChangeSelectionEvent - Selection Changed

Class CgiStudioControl::ChangeStateEvent

ChangeStateEvent - State Change Triggered

Class CgiStudioControl::ChangeValueEvent

ChangeValueEvent - Value Change Triggered

Class CgiStudioControl::ClickEvent

ClickEvent - Clicked

Class CgiStudioControl::ControlTimer::Event

ControlTimer::Event - Timer Expired

Class CgiStudioControl::DragDropEvent

DragDropEvent - Drag and Drop

Class CgiStudioControl::EnterExitEvent

OnEnterExitEvent- On Enter Exit Edit

Class CgiStudioControl::HoverEvent

HoverEvent - Hovered

Member CgiStudioControl::IntervalSwitchBehavior::OnEvent (const FeatStd::Event &event, Candera::EventDispatchResult &dispatchResult)

to handle.

Class CgiStudioControl::IntervalVisualizationEvent

IntervalVisualizationEvent - Interval State Changed

Class CgiStudioControl::PoiUpdateEvent

PoiUpdateEvent - Poi update

Class CgiStudioControl::PressedEvent

PressedEvent - Pressed

Class CgiStudioControl::RegisterSelectionEvent

RegisterSelectionEvent - Register Selection

Class CgiStudioControl::ScrollableDirectionChangedEvent

ScrollableDirectionChangedEvent - Scroll Direction Changed

Class CgiStudioControl::StateEvent

StateEvent - State Changed

Class CgiStudioControl::TrackingEvent

TrackingEvent - Tracking event

Class CgiStudioControl::ValueChangedEvent

ValueChangedEvent - Value Changed

Class CgiStudioControl::VideoControlEvent

VideoControlEvent - Video Control Event

Class CgiStudioControl::VideoStateEvent

VideoStateEvent - Video changed

Revision #6
Created 2023-02-22 09:25:11 UTC by Kanai Tomoaki
Updated 2023-06-19 02:25:01 UTC by Tsuyoshi.Kato