

Use Case Realization Report

UCR-03.02 Refines Event



**Sandia
National
Laboratories**

generated: Apr 13, 2015 10:44 AM

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.



**U.S. DEPARTMENT OF
ENERGY**

1	Use Case Description	3
2	Architecture Description	3
3	Use Case Diagram.....	4
4	Class Diagrams	4
4.1	Classes - Refines Event Display.....	4
4.2	Classes - Refines Event Display - Sub-displays.....	5
4.3	Classes - Event History Display.....	5
4.4	Classes - Event	6
4.5	Classes - Processing Stages.....	6
4.6	Classes - Association Conflict Checker.....	7
4.7	Classes - Signal Detection Template	7
5	Class Descriptions.....	7
6	Sequence Diagrams.....	10
6.1	Flow Overview	10
6.2	Main Flow - Refines Event	10
6.2.1	Operation Descriptions	11
6.3	Expansion Flow - Event - Get Event Hypothesis To Start Refinement	12
6.3.1	Operation Descriptions	12
6.4	Expansion Flow - Refines Event Display - Open Event.....	13
6.4.1	Operation Descriptions	13
6.5	Expansion Flow - Analyst Works Event.....	15
6.5.1	Operation Descriptions	15
6.6	Expansion Flow - Analyst Selects Different Event Hypothesis.....	16
6.6.1	Operation Descriptions	16
6.7	Expansion Flow - Analyst Saves Event Hypothesis.....	17
6.7.1	Operation Descriptions	17
6.8	Expansion Flow - Analyst Updates Comment for Event Hypothesis	18
6.8.1	Operation Descriptions	18
6.9	Expansion Flow - Analyst Sets Preferred Hypothesis for Event	19
6.9.1	Operation Descriptions	19
6.10	Expansion Flow - Analyst Marks Event as Reference Event.....	20
6.10.1	Operation Descriptions.....	20
6.11	Expansion Flow - Analyst Rejects Event.....	21
6.11.1	Operation Descriptions.....	21
6.12	Expansion Flow - Analyst Creates Signal Detection Template	22
6.12.1	Operation Descriptions.....	22
6.13	Expansion Flow - Analyst Copies Event.....	23
6.13.1	Operation Descriptions.....	23
6.14	Expansion Flow - Analyst Refreshes Displayed Data	24
6.14.1	Operation Descriptions.....	24
6.15	Alternate Flow - Display Handles OSD Callbacks.....	25
6.15.1	Operation Descriptions.....	25
6.16	Expansion Flow - Association Conflict Checker - Check For Conflicts	27
6.16.1	Operation Descriptions.....	27
6.17	Expansion Flow - Refines Event Display - Close Event.....	28
6.17.1	Operation Descriptions.....	28
7	State Machine Diagrams	29
8	SSD Mappings	29
9	Notes	35
10	Open Issues.....	36
11	Change History	36

1 Use Case Description

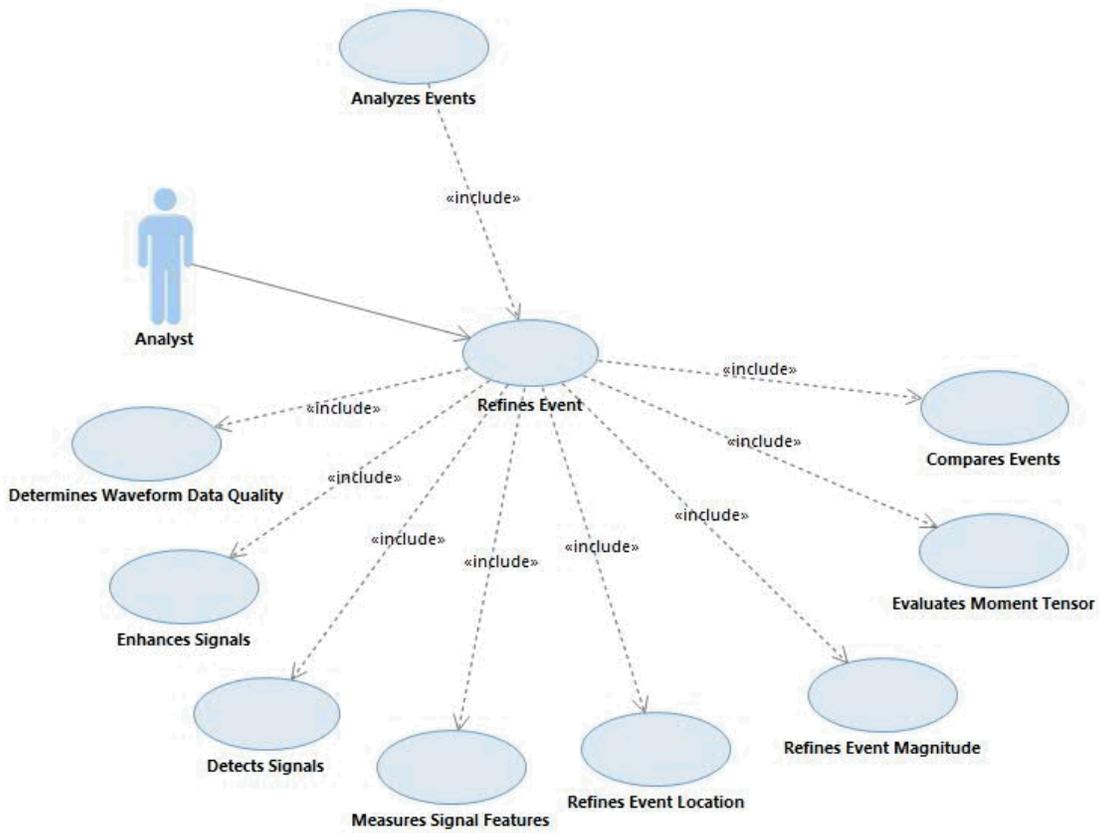
This architecturally significant use case describes how the Analyst refines an event hypothesis. The Analyst checks waveform quality (see 'Determines Waveform Data Quality' UC). For waveforms of sufficient quality, the Analyst enhances signals and suppresses noise on waveforms for relevant stations (see 'Enhances Signals' UC), adds and associates missing detections, and modifies or unassociates detections already associated with the event hypothesis (see 'Detects Signals' UC). The Analyst rejects event hypotheses that are invalid. For valid event hypotheses, the Analyst measures signal features associated with the detections (see 'Measures Signal Features' UC) and evaluates the moment tensor ('Evaluates Moment Tensor' UC). The Analyst uses these signal features to refine the location (see 'Refines Event Location' UC) and magnitude (see 'Refines Event Magnitude' UC) of the event hypothesis. The Analyst compares events to determine how similar events were constructed (see 'Compares Events' UC). The Analyst repeats these steps until satisfied with the results. Analysts may provide feedback for previous Analysts during any of these steps (see 'Provides Analyst Feedback' UC).

This use case is architecturally significant because it captures the interplay between all of the Analyst activities.

2 Architecture Description

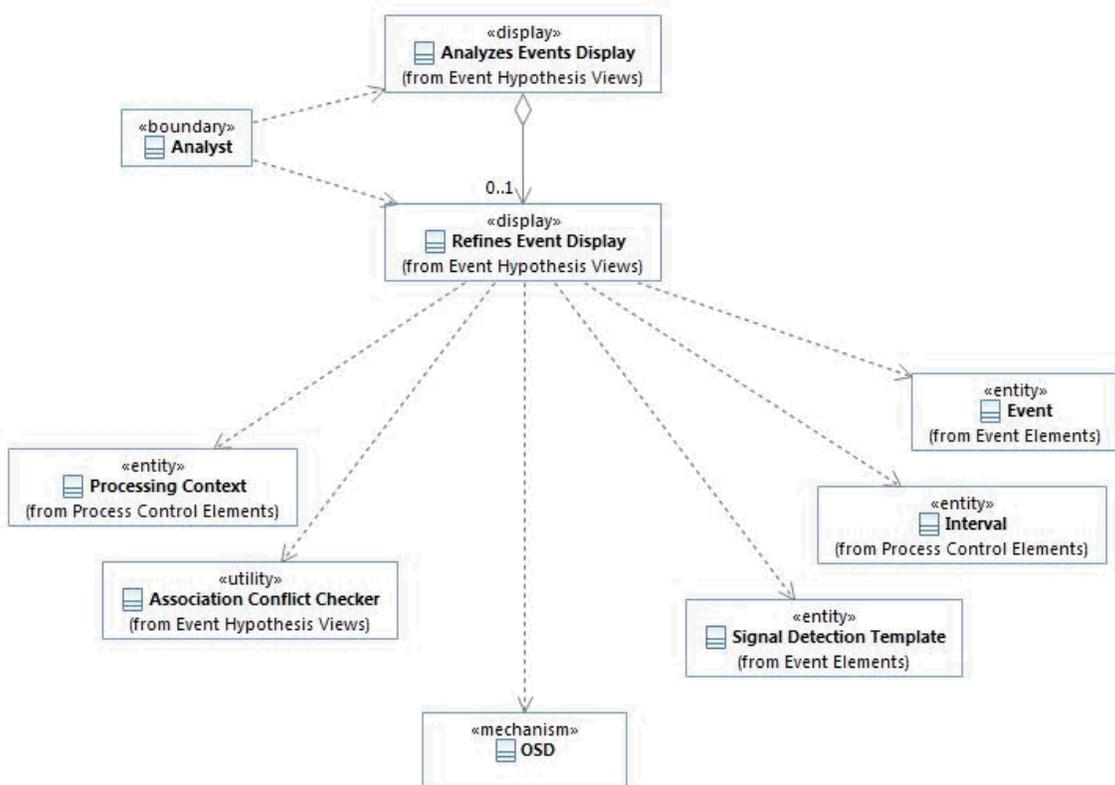
The Analyst refines an event by selecting an event on the Analyzes Events Display to open the Refines Event Display for the event. The Refines Event Display retrieves the latest event hypothesis for the event in the current processing stage (or the preferred hypothesis from previous stage if no hypothesis exists in the current stage) to use as a starting point, creates a local copy of it for the current processing stage, and provides the Analyst with the ability to refine it (depicted in included use cases). As the Analyst refines the event the Event Hypothesis is updated and stored transiently in a private context via the OSD mechanism to make it available to the Processing Sequence Control mechanism for further automatic processing (the Processing Sequence Control mechanism is described in "System Detects Event" UCR). To save their changes, the Analyst selects to save the Event Hypothesis, which the display handles by storing the Event Hypothesis in a global context to persistent storage via the OSD.

3 Use Case Diagram



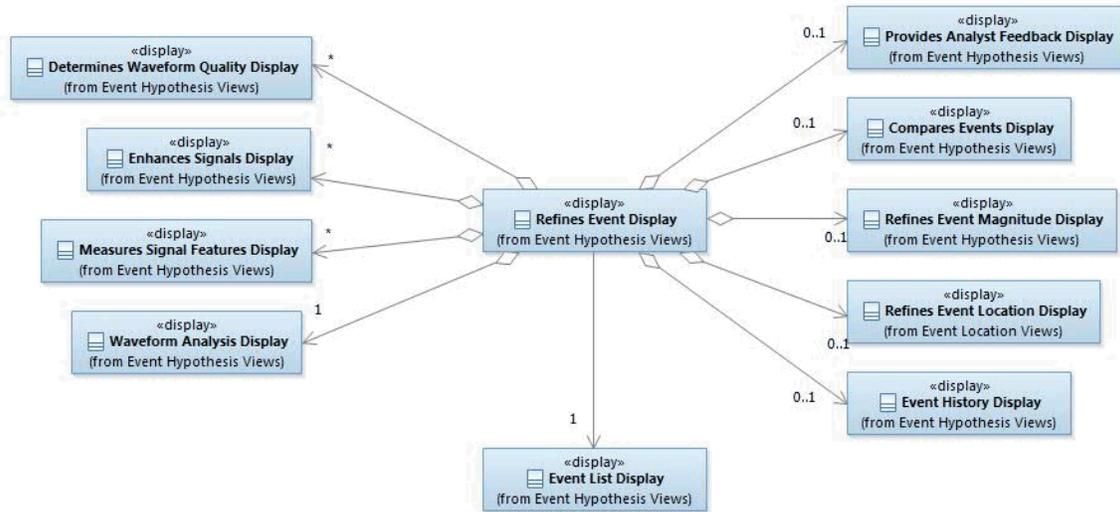
4 Class Diagrams

4.1 Classes - Refines Event Display



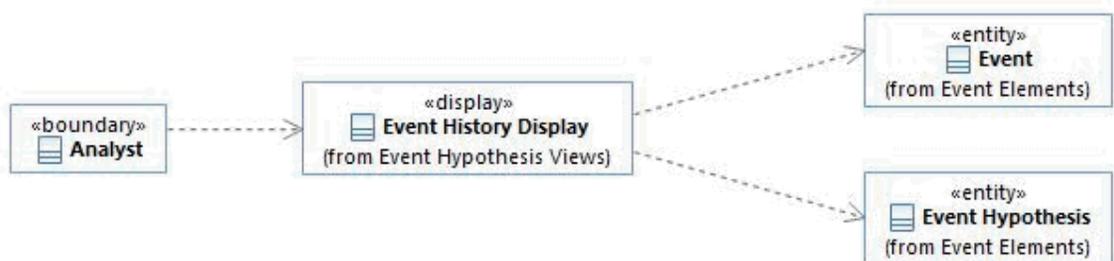
This diagram shows the Refines Event Display and related classes pertaining to this realization. The Analyst opens the Refines Event Display from the Analyzes Events Display. This display subscribes for the Event being refined in order to warn the Analyst if the event is under active review by another Analyst in the same interval. The display subscribes for Intervals in order to warn the Analyst if the interval containing the event is under active review by another Analyst (see "Scans Waveforms and Unassociated Detections" UCR). The Refines Event Display uses the Association Conflict Checker class to check for association conflicts with other events whenever the current Event is modified or another Event in the interval is saved. The display also provides the Analyst with the ability to create Signal Detection Templates, which it stores in the OSD. Refines Event Display stores the refined Event in the OSD.

4.2 Classes - Refines Event Display - Sub-displays



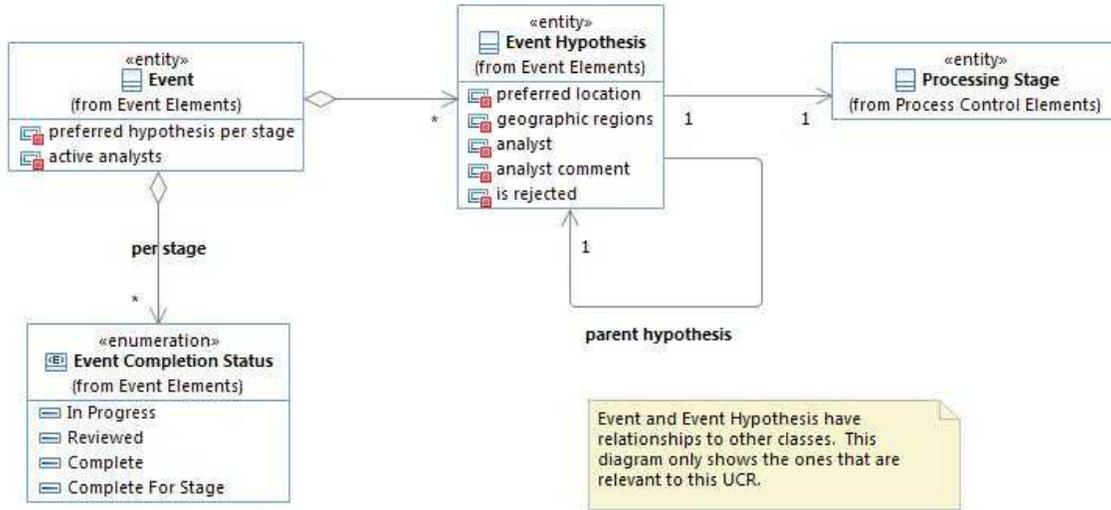
This class shows sub-displays of the Refines Event Display. The Refines Event Display creates and manages these sub-displays based on Analyst actions. Analyst interactions with these sub-displays are described in the corresponding UCRs; however, in general, the OSD mechanism is used to synchronize information between the displays. When the Analyst first opens the event, the Refines Event Display creates a new Event Hypothesis and stores it in the OSD (in a private context, not visible to other Analysts). The Refines Event Display then subscribes for changes to this Event Hypothesis via the OSD. As the Analyst interacts with the various sub-displays, those Analyst actions may trigger processing on the privately stored Event Hypothesis; however, the sub-displays do not have knowledge of which processing steps will be performed since the processing sequences to be executed in response to Analyst actions are configurable (see "Defines Processing Sequence" UCR) and known only by the Processing Sequence Control mechanism. The Refines Event Display is informed of any processing performed on the Event Hypothesis via OSD callbacks.

4.3 Classes - Event History Display



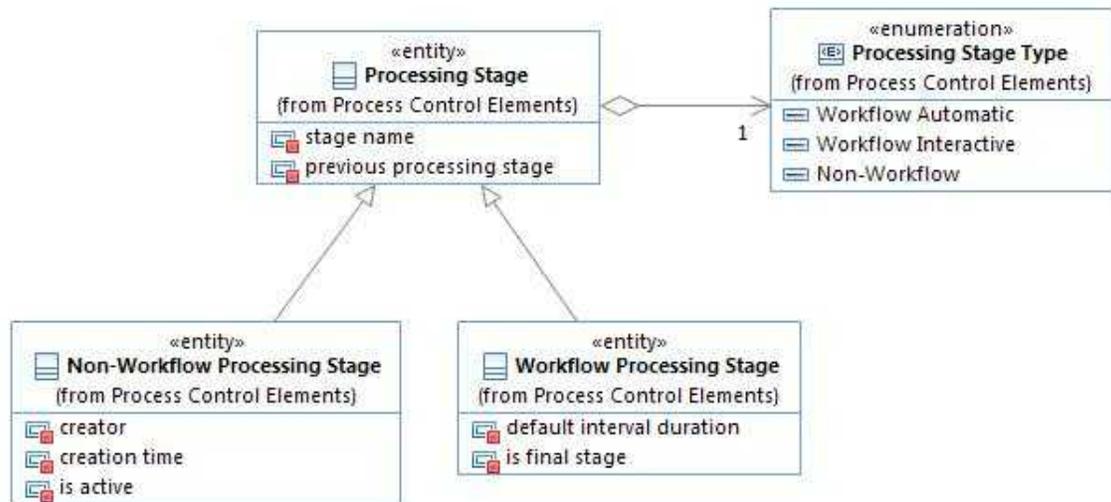
This diagram shows the Event History Display and related classes.

4.4 Classes - Event



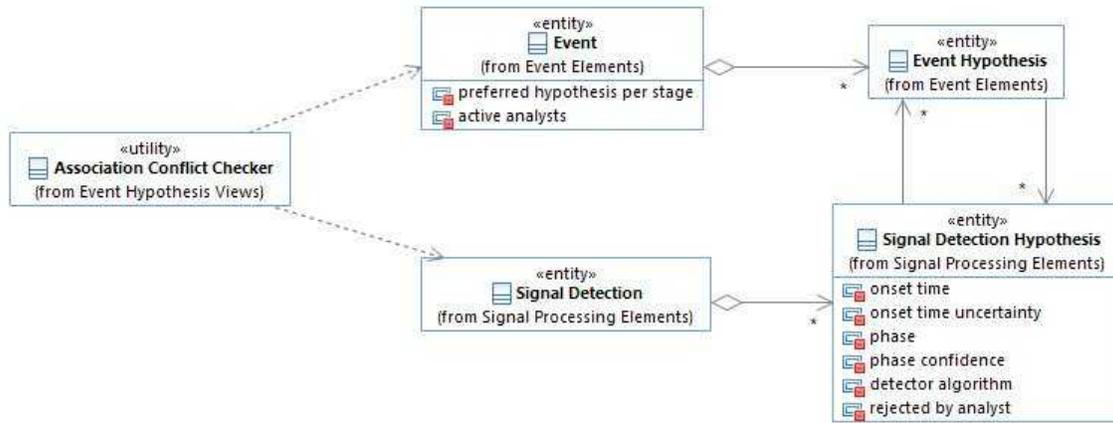
This diagram shows details of the Event and Event Hypothesis classes relevant to this UCR. Refinement of an Event results in a new Event Hypothesis. The Analyst potentially creates multiple Event Hypotheses for a given Event during a single processing stage, and designates one of them as the "preferred" hypotheses for the event for that stage (each stage can have a different preferred hypothesis for the event). Each Event also has an Event Completion Status, which reflects the Analyst's determination of the level of completeness of the event within the stage. The Analyst specifies an Event Completion Status of "In Progress", "Reviewed" or "Complete" when saving the event. The transition to "Complete For Stage" is covered in a separate UCR (see "Marks Processing Stage Complete" UCR).

4.5 Classes - Processing Stages



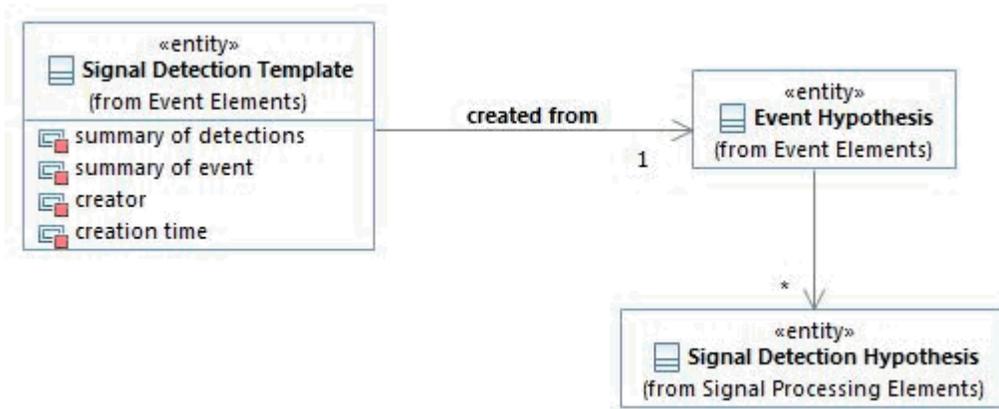
This diagram shows the Processing Stage class hierarchy and how Non-Workflow Processing Stages fit into it. Processing results (e.g. event and signal hypotheses) are linked to the Processing Stage in which they were created, which may be any of the three types of Processing Stage (Interactive, Automatic or Non-Workflow). Workflow Processing Stages are defined by the System Maintainer as part of the overall Processing Configuration (see "Defines Processing Sequence" UCR), whereas Non-Workflow Processing Stages are defined by individual Analysts (see "Alternate Flow - Analyst Defines Non-Workflow Processing Stage" in this UCR).

4.6 Classes - Association Conflict Checker



This diagram shows details of the Association Conflict Checker class. The class retrieves Events and Signal Detections from the OSD and checks for the case where more than one Event has a preferred Event Hypothesis for the stage that is associated to a Signal Detection Hypothesis for the same Signal Detection.

4.7 Classes - Signal Detection Template



This diagram shows details for the Signal Detection Template class. The Analyst may create a Signal Detection Template via the Refines Event Display based on the Signal Detection Hypotheses associated to the current Event Hypothesis.

5 Class Descriptions

<<boundary>> Analyst

Represents the Analyst actor.

<<display>> Analyzes Events Display

Display that provides the Analyst with the ability to analyze data within a specified time interval in order to find or refine events.

<<display>> Refines Event Display

Display that provides the Analyst with the ability to refine an event. Each saved refinement of the event results in a new Event Hypothesis.

<<display>> Refines Event Location Display

Provides the Analyst with ability to enter event location parameters and initiate computation of a location for an event hypothesis.

<<display>> *Waveform Analysis Display*

Displays a set of waveforms and provides the Analyst with the ability to interact with them (e.g. create/modify/reject signal detections, associate/disassociate detections and events).

<<entity>> *Event*

Represents information about an Event. Keeps track of all the Event Hypotheses for the event, which hypothesis is the preferred one for each processing stage, the active analysts for the event (i.e. whether the event is under "active review"), whether the event is "complete" for each processing stage, and other event-related information.

<<entity>> *Event Hypothesis*

Represents geophysical information about an Event as determined by an Analyst or through pipeline processing. There can be multiple hypotheses of the same Event (e.g. different associated signal detection hypotheses, different location solutions).

<<entity>> *Interval*

Class for tracking the status of interactive or automatic processing on a specific timeframe of data. Specialized intervals exist for Processing Stage, Processing Activity, and Processing Sequence.

<<entity>> *Non-Workflow Processing Stage*

Represents a Processing Stage that is not part of the System Maintainer-defined Analyst workflow. Analysts may define such stages at any time to store their Processing Results, but the stored results in such stages are not used by Analysts or the system during workflow processing.

<<entity>> *Processing Context*

Represents the context in which data is being stored and/or processed. This includes the processing session (e.g. processed by Analyst vs. processed by System). For Analyst processing, may identify the Analyst work session. For System processing, may identify the Processing Sequence and/or Processing Unit being executed (including a way to identify a particular Processing Sequence and Processing Unit among the many possible instantiations), the visibility for the results (private vs. global), and the lifespan of the data (transient vs. persistent). This information is needed by the Processing Sequence Control to manage the execution of Processing Sequences, which may execute in the context of an Analyst refining an event or in the context of the system initiating automatic processing. It is also needed by the Object Storage and Distribution (OSD) mechanism to determine how to store and distribute the data.

<<entity>> *Processing Stage*

Represents a named stage of data processing, which may be part of the System Maintainer-defined workflow or an Analyst-defined stage outside the workflow. All Processing Results are associated to a Processing Stage. The "previous processing stage" attribute is indicates the stage to be used by as the default starting point when creating new processing results in the stage (e.g. when refining an event in the stage). Note that this attribute is optional, since some stages such as the one for the Duty Officer do not have the concept of a default starting point.

<<entity>> *Signal Detection*

Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. For an unassociated Signal Detection the preferred hypothesis is the most recently created hypothesis. For an associated Signal Detection the preferred hypothesis is the one associated to a preferred Event Hypothesis.

<<entity>> *Signal Detection Hypothesis*

Represents geophysical information about a Signal Detection as determined by an Analyst or through pipeline

processing. There can be multiple hypotheses of the same Signal Detection (e.g. different onset times, different phase labels).

<<entity>> *Signal Detection Template*

A template that represents the pattern of signal detections for an event (i.e. channels detected, relative positions for each detection, phases, etc.). An Analyst may apply the template to quickly build new events that match the pattern of detections. Also includes summary information about the original event from which the template was created (e.g. event location, magnitude, etc.), as an aid to the Analyst in finding and applying a relevant template.

<<entity>> *Workflow Processing Stage*

Represents a Processing Stage that is part of the System Maintainer-defined Analyst workflow.

<<mechanism>> *OSD*

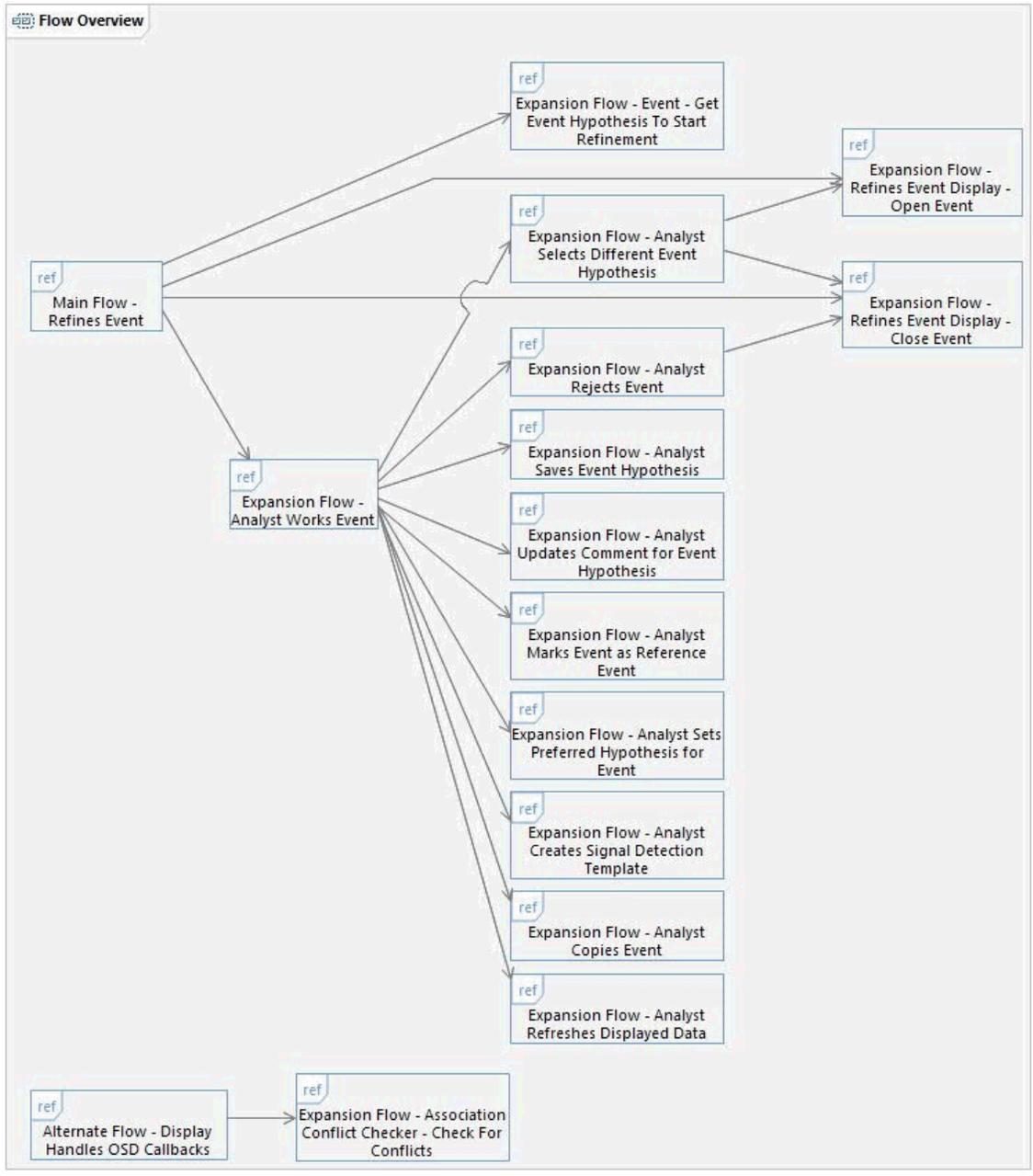
Represents the Object Storage and Distribution mechanism for storing and distributing data objects internally within the system.

<<utility>> *Association Conflict Checker*

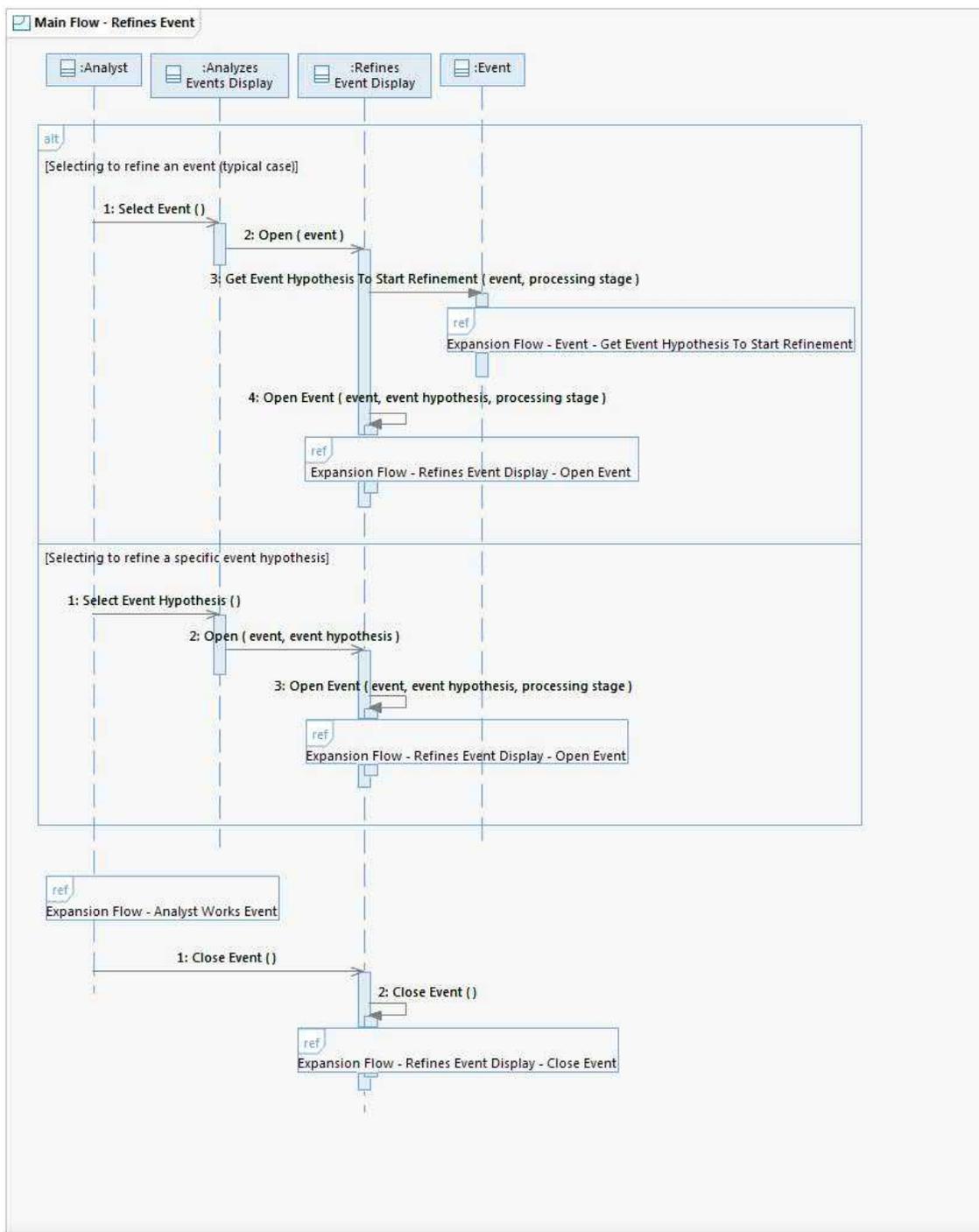
Utility class for checking that preferred event hypotheses within a processing stage do not share any signal detections.

6 Sequence Diagrams

6.1 Flow Overview



6.2 Main Flow - Refines Event



This flow shows the main flow for refining an event. The Refines Event Display is typically opened with an event, in which case the display automatically determines which event hypothesis to use as a starting point (the Analyst can select a different hypothesis as a starting point - see "Expansion Flow - Analyst Selects Different Event Hypothesis"). The Analyst may also open the Refines Event Display with a specific event hypothesis to refine. In this case the display uses that hypothesis as the starting point.

6.2.1 Operation Descriptions

Operation: Refines Event Display::Open Event()

Open the given event for refinement in the current processing stage, using the given event hypothesis as a starting point.

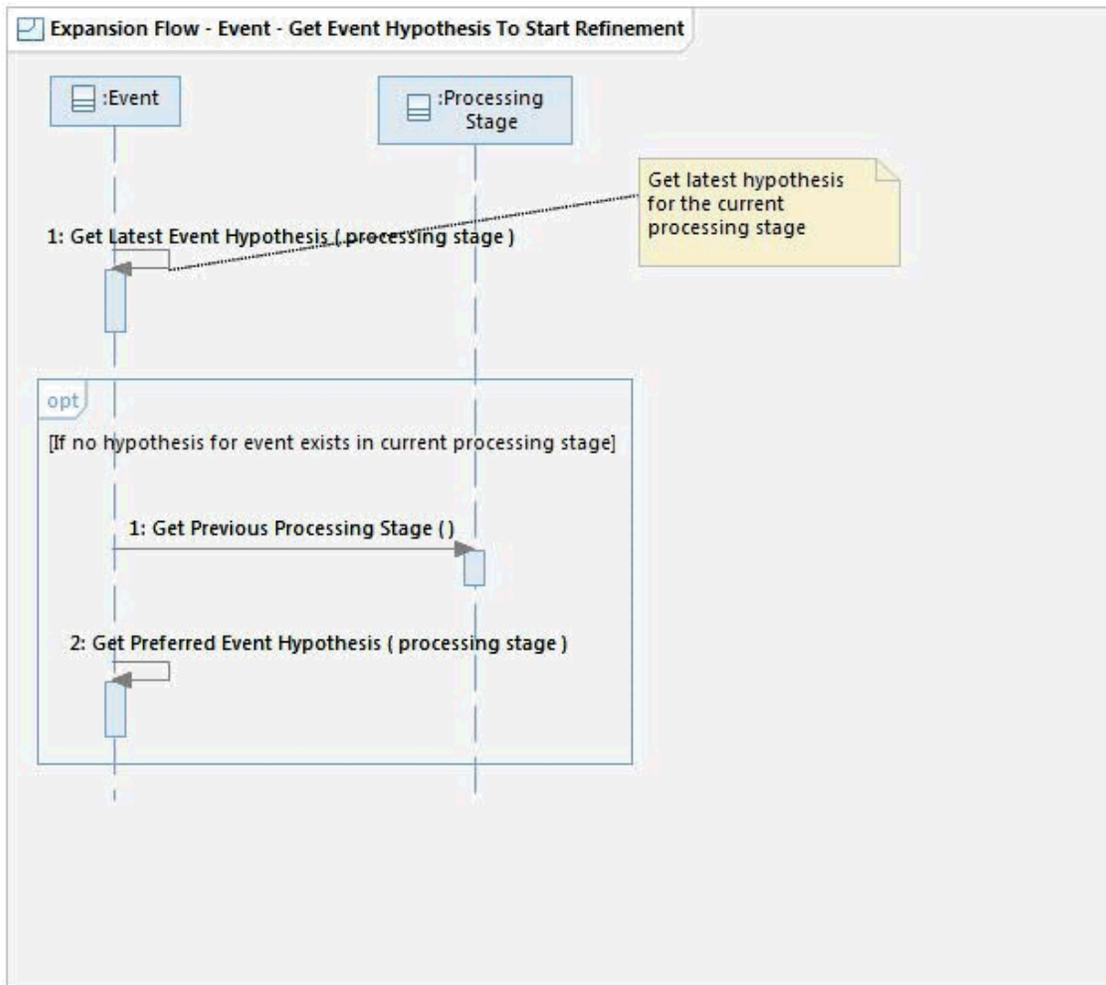
Operation: Event::Get Event Hypothesis To Start Refinement()

Returns an event hypothesis to use as a starting point for refining the event in the given processing stage. If the processing stage is an Interactive Processing Stage, looks in the previous stage for a preferred hypothesis in that stage. If none exists, continues search in next previous stage, etc. until a preferred hypothesis is found. Note that the stage order is defined in the Processing Configuration class, which is defined by the System Maintainer. If the processing stage is a non-workflow Processing Stage

Operation: Refines Event Display::Open Event()

Open the given event for refinement in the current processing stage, using the given event hypothesis as a starting point.

6.3 Expansion Flow - Event - Get Event Hypothesis To Start Refinement

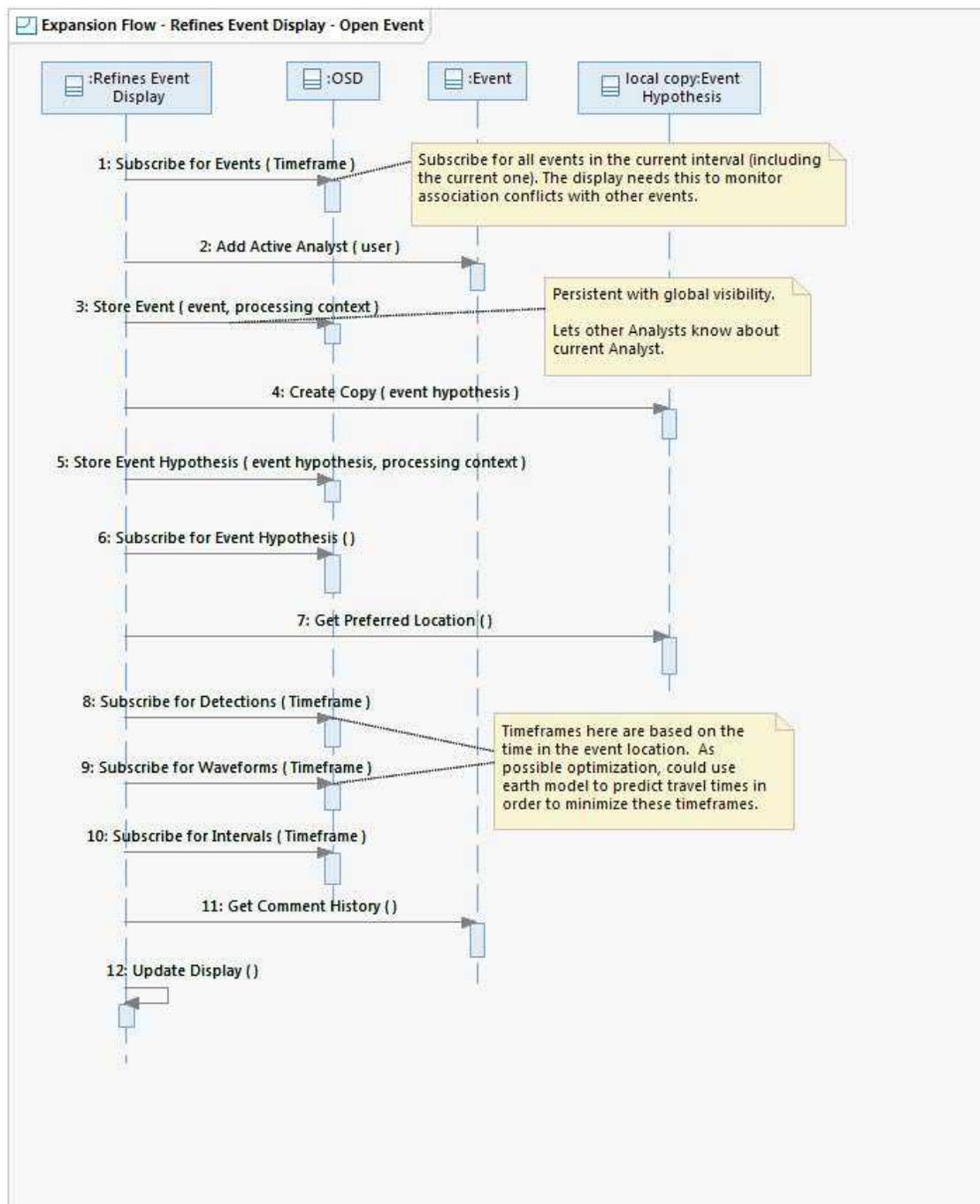


This flow shows how the Event class finds an event hypothesis to use as a starting point for refinement in the current stage, which may be a workflow or non-workflow stage. Note that, in the case of a workflow stage the "previous processing stage" is configured by the System Maintainer (see "Defines Processing Sequence" UCR) whereas in the case of a non-workflow stage the "previous processing stage" is set by the Analyst when they define the stage (see "Selects Data for Analysis" UCR).

6.3.1 Operation Descriptions

None

6.4 Expansion Flow - Refines Event Display - Open Event



This flow shows how the Refines Event Display opens an event for refinement. The event hypothesis to use as a starting point is an input to this flow. The system keeps track of all the analysts that are working an event ("active analysts") and warns if the event is under active review by another analyst or overlaps an interval that is under active review by another analyst (see "Alternate Flow - Display Handles OSD Callbacks"). The display creates a new Event Hypothesis instance based on passed-in event hypothesis. The display subscribes for detections and waveforms around the event in order to display them. The comment history is also displayed.

6.4.1 Operation Descriptions

Operation: `OSD::Subscribe for Events()`

Subscribes for changes to Event objects within the given timeframe. Callbacks are invoked on subscribers any time an Event within the timeframe is added or modified.

Operation: OSD::Subscribe for Detections()

Subscribes for updates regarding signal detection creations, modifications, and associations occurring within the specified timeframe. This includes updates for new or modified unassociated signal detections.

Operation: OSD::Subscribe for Waveforms()

Subscribes for updates regarding raw and derived waveforms occurring within a specified timeframe. This includes information about what waveforms have been acquired by the System as well as what derived waveforms have been formed, but does not include the actual waveform data.

Operation: Event::Add Active Analyst()

Add the given Analyst to the set of active Analysts for the event. If the event has active analysts it is said to be under "active review".

Operation: OSD::Store Event()

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: Event Hypothesis::Create Copy()

Create a copy of the given event hypothesis. The copy has all of the same information as the original (e.g. same detections, location, etc.), with the following exceptions:

- The copy points to the original as its parent
- The copy starts out with an empty Analyst comment

Operation: OSD::Store Event Hypothesis()

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: Refines Event Display::Update Display()

Update the display of the Event Hypothesis that is currently being refined to reflect any changes that may have occurred. Indicate items that are out-of-date or inconsistent (e.g. beam may be out-of-date after refining event location).

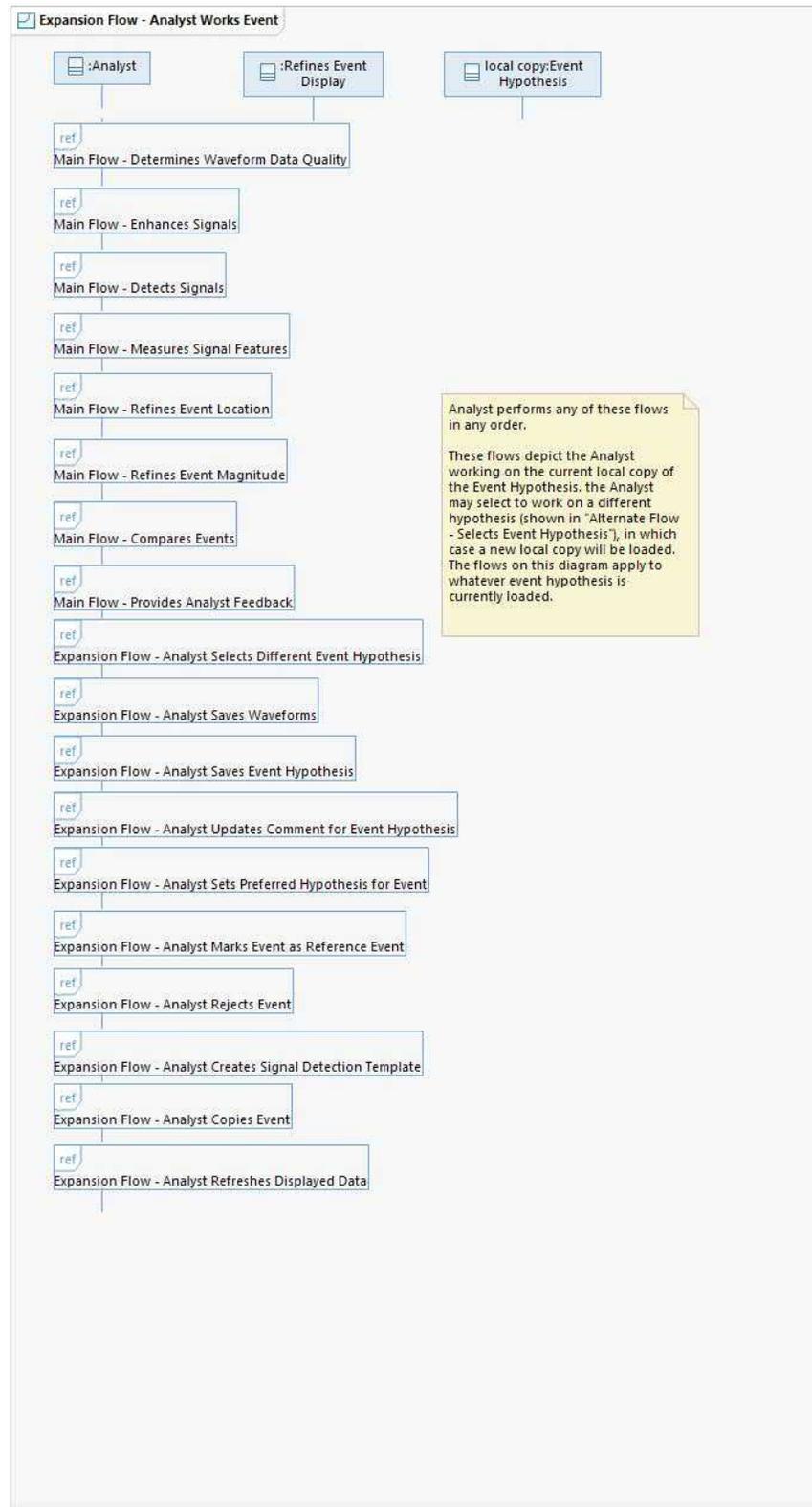
Operation: Event::Get Comment History()

Return all Analyst-entered comments associated with the Event.

Operation: OSD::Subscribe for Intervals()

Subscribes for changes to Interval objects that overlap with the given timeframe. Interval objects track the active analysts and completion status of intervals corresponding to processing stages and processing activities within processing stages. Callbacks are invoked on subscribers any time the set of active analysts or completion status for an Interval changes.

6.5 Expansion Flow - Analyst Works Event

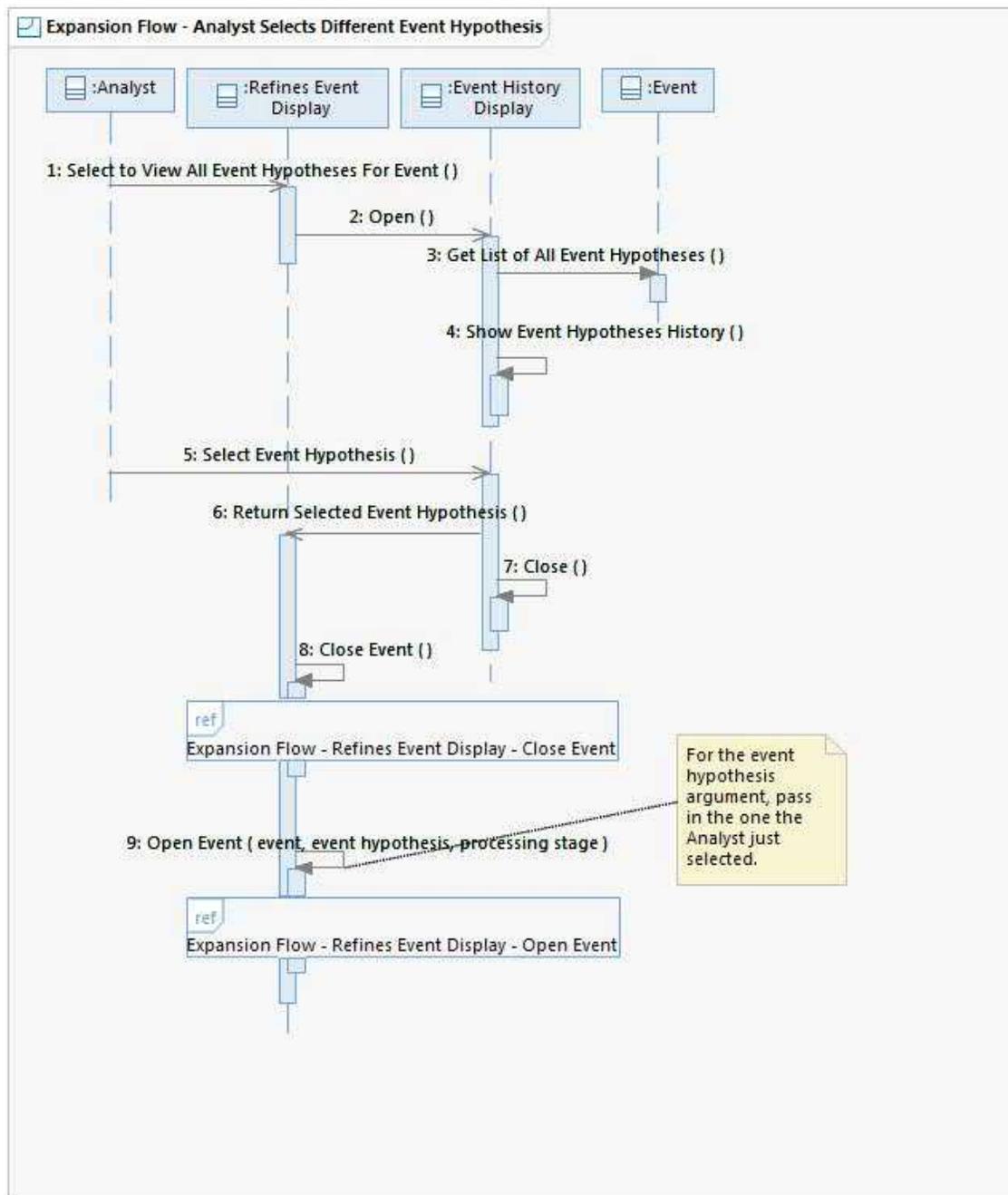


This flow shows the actions an Analyst can perform as part of refining an event.

6.5.1 Operation Descriptions

None

6.6 Expansion Flow - Analyst Selects Different Event Hypothesis



This flow shows the Analyst selecting to refine a different Event Hypothesis for the event (other than the current one).

6.6.1 Operation Descriptions

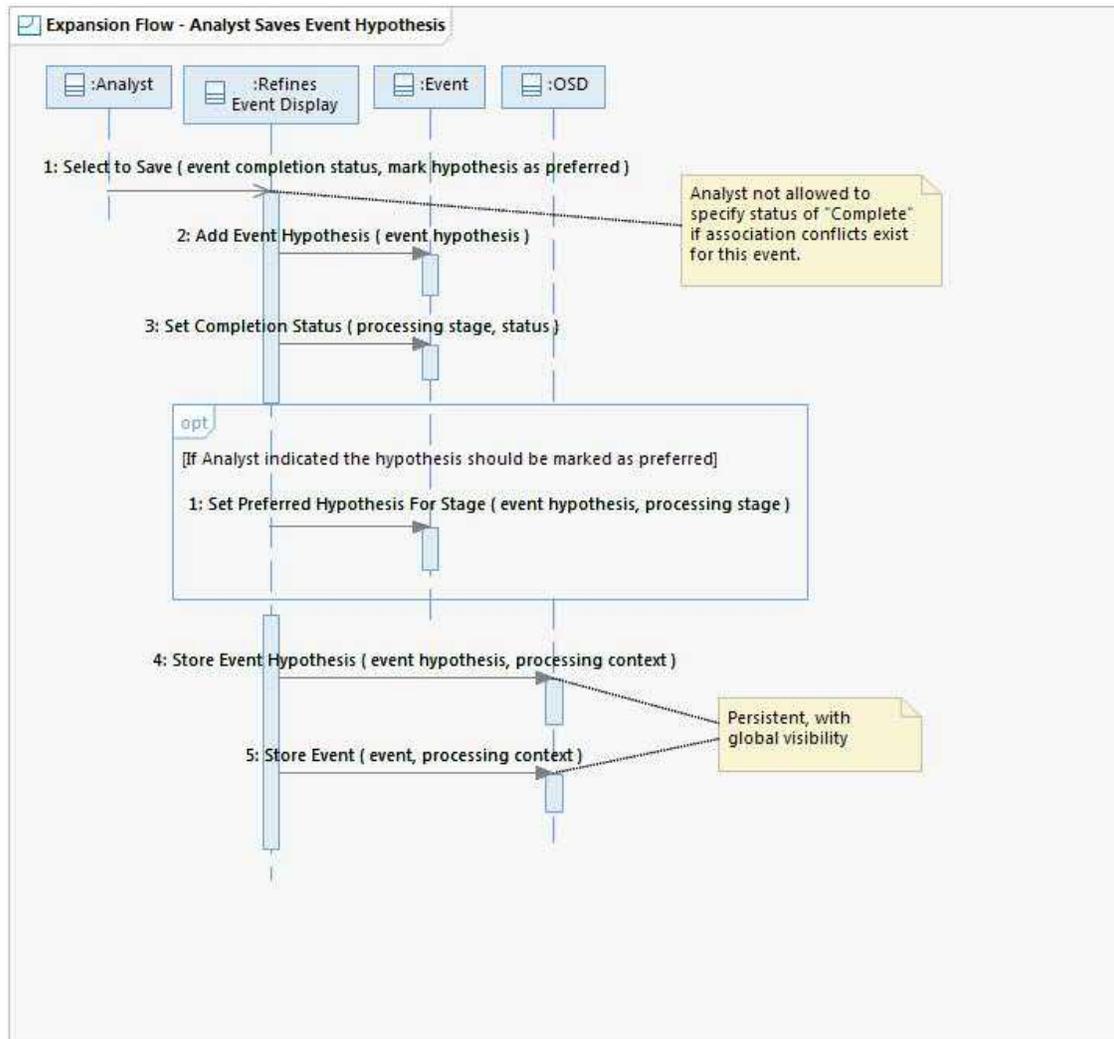
Operation: Refines Event Display::Open Event()

Open the given event for refinement in the current processing stage, using the given event hypothesis as a starting point.

Operation: Event::Get List of All Event Hypotheses()

Return a list of all the Event Hypothesis for the given event, including summary information such as the processing stage and which hypotheses have been designated as preferred.

6.7 Expansion Flow - Analyst Saves Event Hypothesis



This flow shows how the Analyst saves the event hypothesis they are refining to persistent storage and makes it visible to other Analysts. Once the event hypothesis is saved to persistent storage it can never be modified again (but the Analyst can create a new event hypothesis to further refine the event). When saving the event, the Analyst specifies the completion status for the event (see Event Completion Status class on diagram "Classes - Event") and whether to mark the hypothesis as the preferred one for the event.

6.7.1 Operation Descriptions

Operation: *OSD::Store Event Hypothesis()*

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

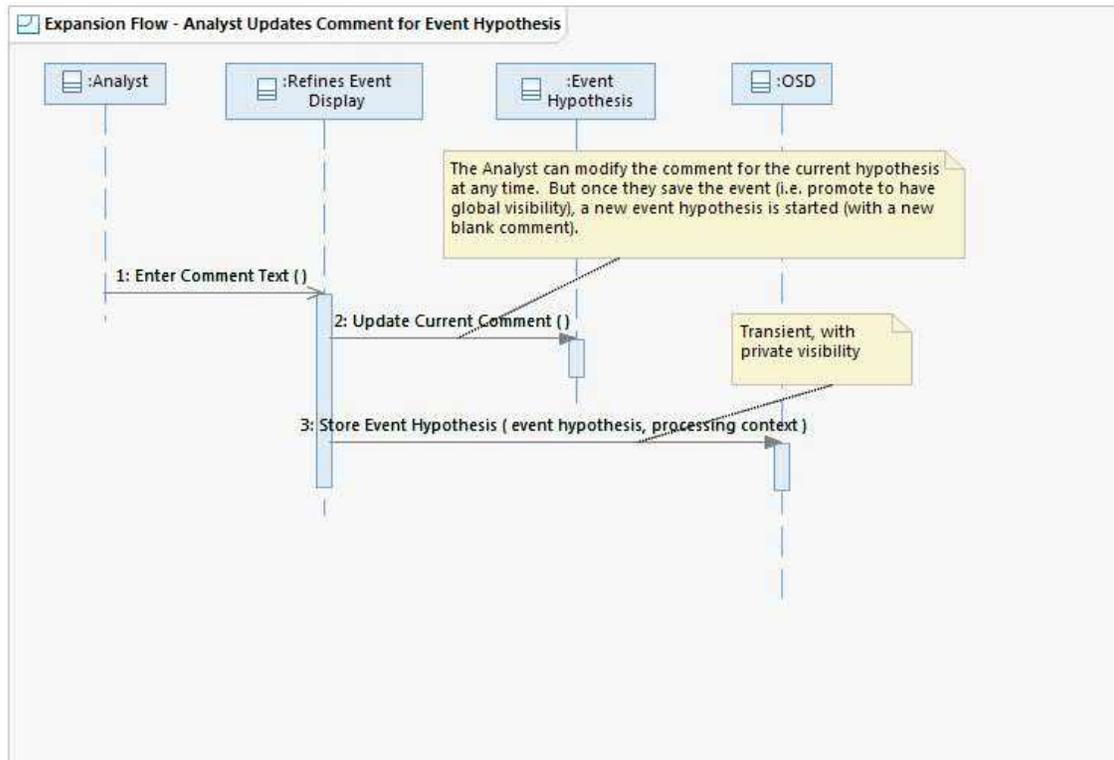
Operation: *OSD::Store Event()*

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: *Event::Set Completion Status()*

Set the Event Completion Status of an event in the given processing stage to the given value.

6.8 Expansion Flow - Analyst Updates Comment for Event Hypothesis



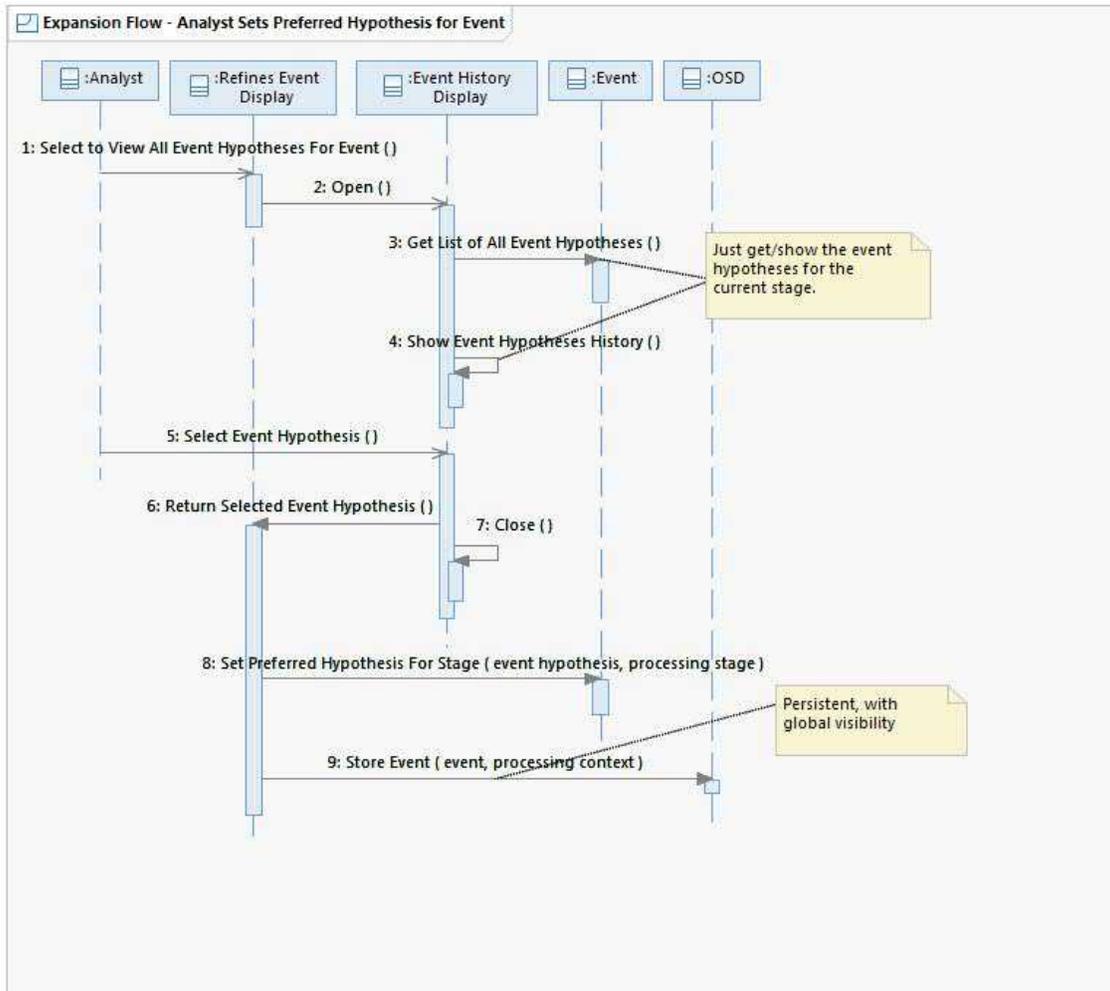
This flow shows how the Analyst updates comments on the Event Hypothesis.

6.8.1 Operation Descriptions

Operation: *OSD::Store Event Hypothesis()*

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

6.9 Expansion Flow - Analyst Sets Preferred Hypothesis for Event



This flow shows the Analyst designating an event hypothesis as preferred for a specified processing stage. Marking an hypothesis as preferred causes the event to be immediately stored in a global context (visible to other analysts), but does not change its Event Completion Status.

6.9.1 Operation Descriptions

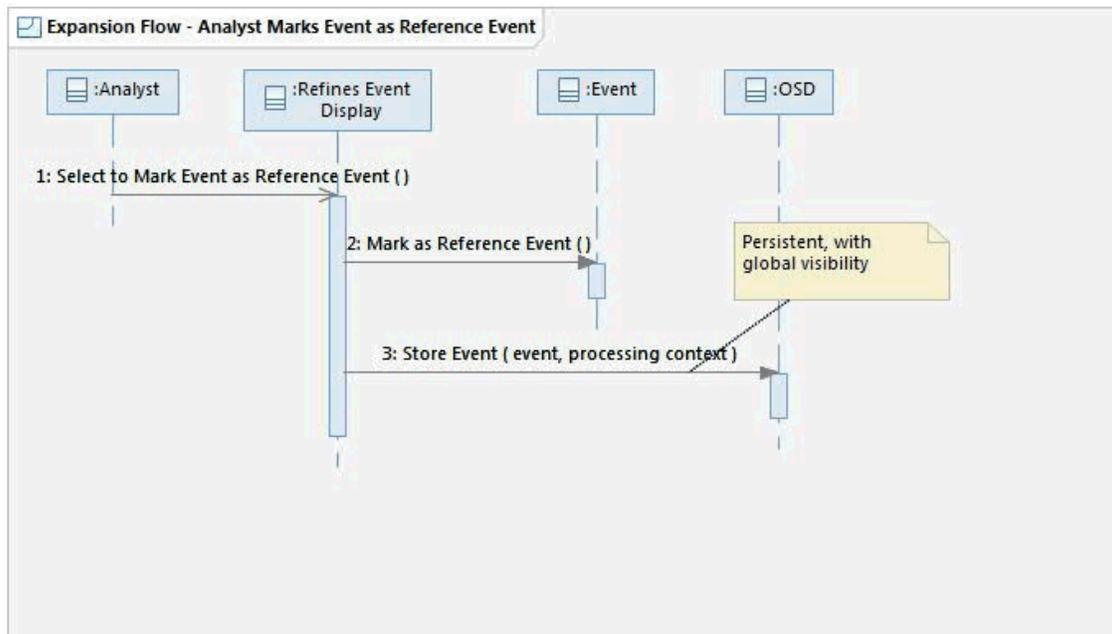
Operation: OSD::Store Event()

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: Event::Get List of All Event Hypotheses()

Return a list of all the Event Hypothesis for the given event, including summary information such as the processing stage and which hypotheses have been designated as preferred.

6.10 Expansion Flow - Analyst Marks Event as Reference Event



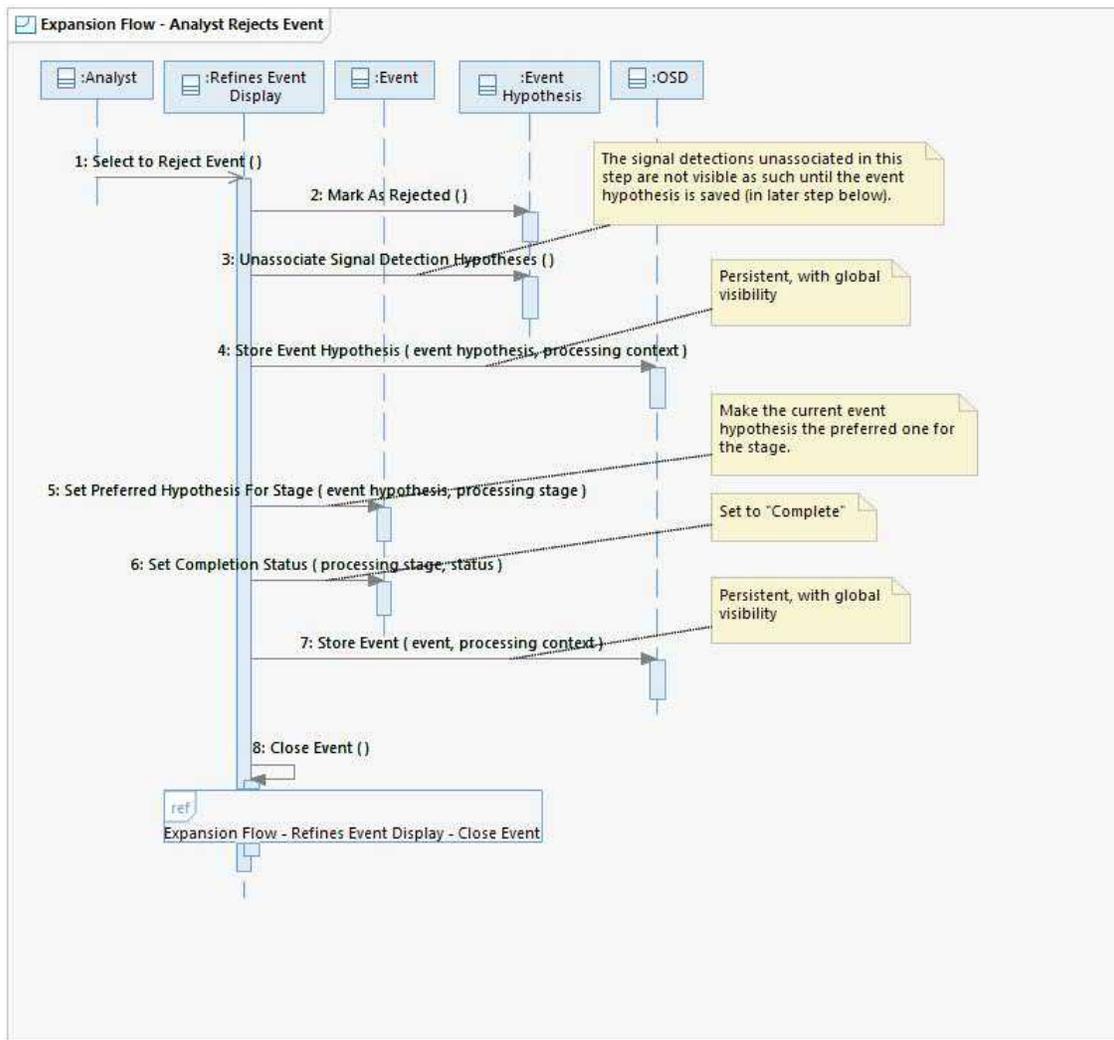
This flow shows the Analyst marking an event as a reference event. Although not shown, they may also unmark a previously marked event in a similar manner. Note that it is the Event that is marked (not the Event Hypothesis), since the mark logically applies to the Event rather than any specific hypothesis.

6.10.1 Operation Descriptions

Operation: `OSD::Store Event()`

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

6.11 Expansion Flow - Analyst Rejects Event



This flow shows how the Refines Event Display handles rejecting an event. Rejecting an event is accomplished by unassociating all signal detection hypotheses from the current event hypothesis, making the current event hypothesis the preferred hypothesis for the current stage, saving the Event and the Event Hypothesis objects, and closing the Refines Event Display.

6.11.1 Operation Descriptions

Operation: OSD::Store Event Hypothesis()

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

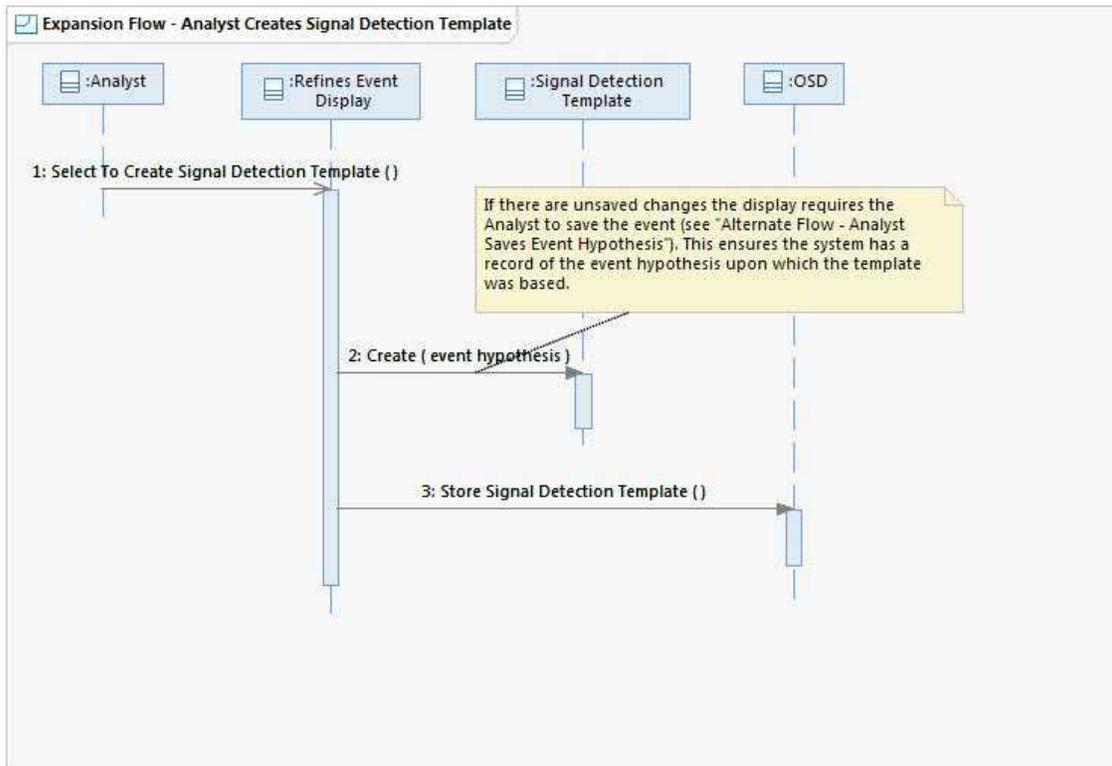
Operation: OSD::Store Event()

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: Event::Set Completion Status()

Set the Event Completion Status of an event in the given processing stage to the given value.

6.12 Expansion Flow - Analyst Creates Signal Detection Template

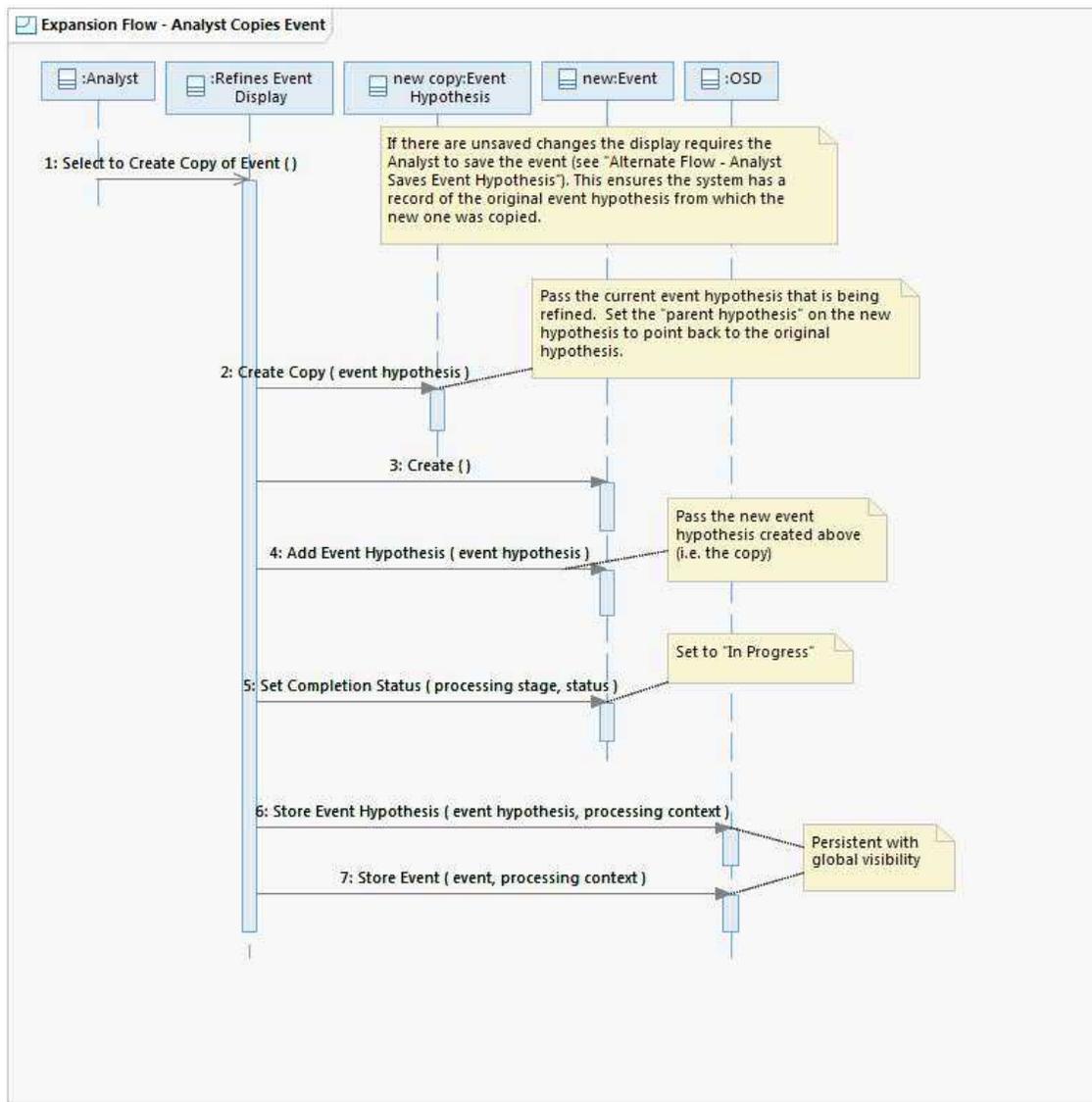


The flow shows the Analyst creating a new Signal Detection Template from the current Event Hypothesis. Stored templates may be applied by the Analyst when building a new event (see "Builds Event" UCR).

6.12.1 Operation Descriptions

None

6.13 Expansion Flow - Analyst Copies Event



This flow shows the Analyst creating a copy of the event they are currently refining. Initially, the copy will share all of the signal detection hypotheses as the original (resulting in conflicts which will be detected by the Association Conflict Checker - see "Alternate Flow - Display Handles OSD Callbacks"). The Analyst will need to refine each event individually to manually remove the conflicts.

6.13.1 Operation Descriptions

Operation: Event Hypothesis::Create Copy()

Create a copy of the given event hypothesis. The copy has all of the same information as the original (e.g. same detections, location, etc.), with the following exceptions:

- The copy points to the original as its parent
- The copy starts out with an empty Analyst comment

Operation: OSD::Store Event Hypothesis()

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

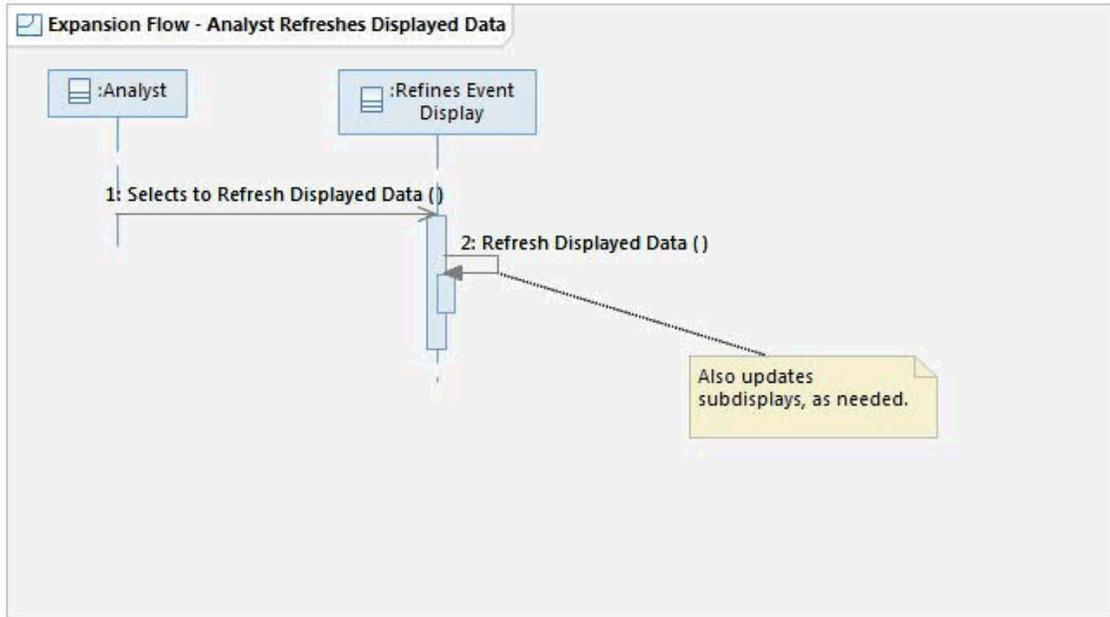
Operation: OSD::Store Event()

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

Operation: Event::Set Completion Status()

Set the Event Completion Status of an event in the given processing stage to the given value.

6.14 Expansion Flow - Analyst Refreshes Displayed Data



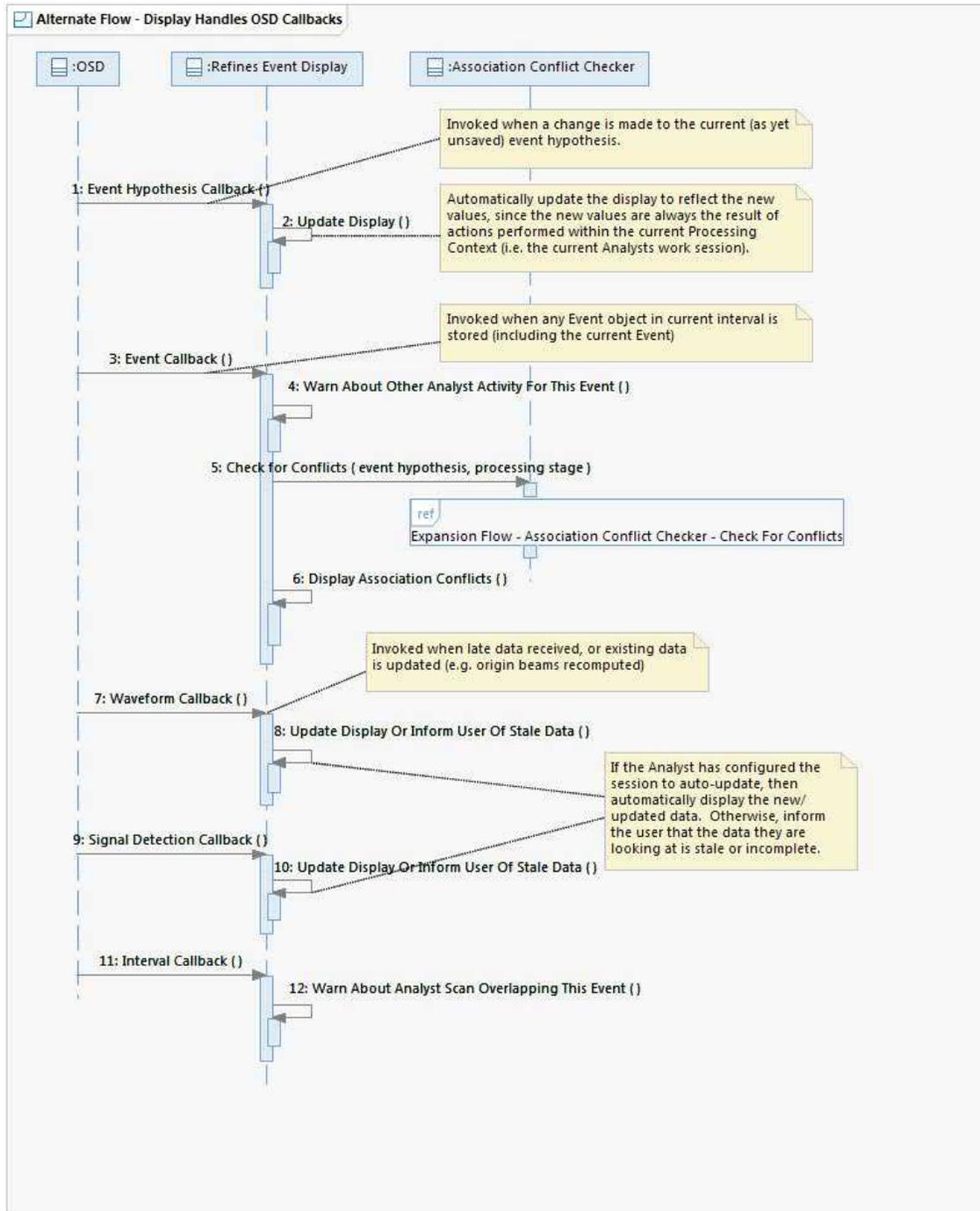
This flow shows how the Analyst refreshes his/her display to show the latest waveforms, signal detections and signal detection associations. The Analyst needs this capability since late-arriving waveforms, signal detections and signal detection associations made by other Analysts are not automatically displayed to the Analyst. The Analyst is notified when there is new data that is not shown on the display (see "Alternate Flow - Display Handles OSD Callbacks"). The Analyst may refresh the display to show that data at any time via this flow. Note that the display does not need to retrieve the new data from the OSD since it already has it due to subscriptions with the OSD (see "Expansion Flow - Refines Event Display - Open Event").

6.14.1 Operation Descriptions

Operation: Refines Event Display::Refresh Displayed Data()

Update displayed waveforms and signal detections to reflect the current state within the processing stage.

6.15 Alternate Flow - Display Handles OSD Callbacks



This flow shows how the Refines Event Display handles various callbacks from the OSD. The Refines Event Display subscribes for the current Event Hypothesis in order to monitor updates to it as a result of executed Processing Sequences. The display subscribes for all Events in the interval in order to monitor other Analyst activity on the current event and to check for association conflicts with other events. The display subscribes for Signal Detections and Waveforms in order to display updates to that information made by other analysts. The display subscribes for Intervals to determine if an interval that overlaps the current event is under active review by another analyst.

6.15.1 Operation Descriptions

Operation: Refines Event Display::Event Callback()

Callback invoked any time there is a change in the subscribed Event (e.g. a new Event Hypothesis for the Event

is saved, or the preferred hypothesis for a processing stage changes).

Operation: Refines Event Display::Warn About Other Analyst Activity For This Event()

Warn the current Analyst about another Analyst working on the current event.

Operation: Refines Event Display::Signal Detection Callback()

Invoked any time the set of signal detections that fall within the current time interval changes. The callback indicates what changed.

Operation: Refines Event Display::Waveform Callback()

Invoked any time new raw or derived waveforms overlapping the time of the event are received (e.g. late data, beams).

Operation: Refines Event Display::Event Hypothesis Callback()

Callback invoked whenever portions of the Event Hypothesis are changed by the system. This callback can only occur as part of automatic processing sequences executed by the Processing Sequence Control mechanism, since changes made by other Analysts are stored in separate Event Hypotheses.

Operation: Refines Event Display::Update Display()

Update the display of the Event Hypothesis that is currently being refined to reflect any changes that may have occurred. Indicate items that are out-of-date or inconsistent (e.g. beam may be out-of-date after refining event location).

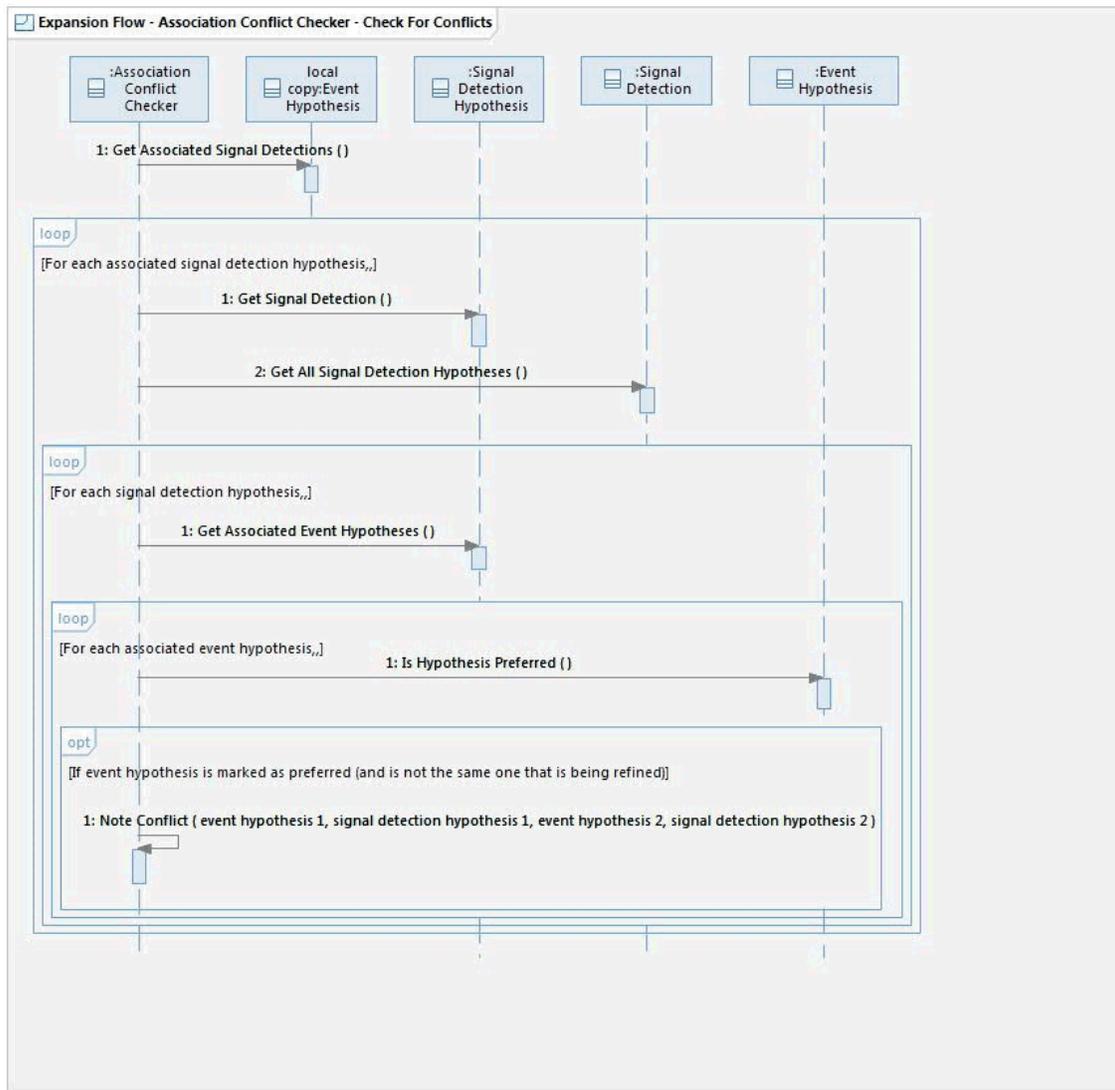
Operation: Association Conflict Checker::Check for Conflicts()

Checks the given event hypothesis against the other events in the processing stage for association conflicts. Only checks for conflicts between event hypotheses marked as preferred. A conflict exists if a signal detection is associated to more than one preferred hypothesis in the processing stage.

Operation: Refines Event Display::Warn About Analyst Scan Overlapping This Event()

Warn the Analyst if the interval that overlaps the current event being refined is under active review by another Analyst.

6.16 Expansion Flow - Association Conflict Checker - Check For Conflicts



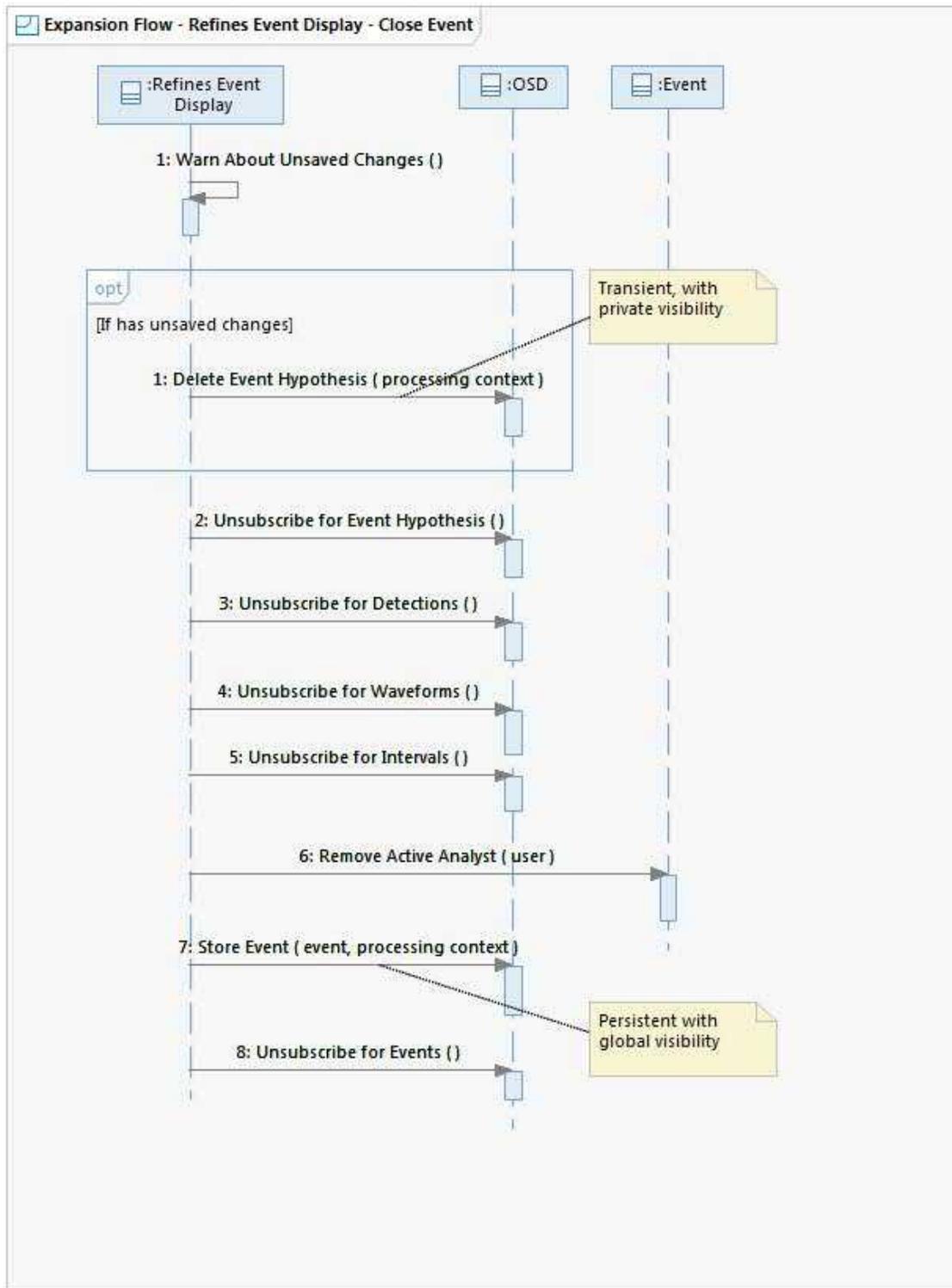
This flow shows how the Association Conflict Checker checks for association conflicts between the local (unsaved) event hypothesis and other event hypotheses. The local event hypothesis is input to this flow. The flow returns a list of association conflicts. Note that, by definition, a conflict can only exist with an event hypothesis that is marked as preferred. Note also that each event can have at most one event hypothesis marked as preferred for each processing stage.

6.16.1 Operation Descriptions

Operation: Event Hypothesis::Get Associated Signal Detections()

Returns the Signal Detection that owns each Signal Detection Hypothesis associated to the Event Hypothesis.

6.17 Expansion Flow - Refines Event Display - Close Event



This flow shows what the Refines Event Display does upon closing the current event.

6.17.1 Operation Descriptions

Operation: `OSD::Store Event()`

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

7 State Machine Diagrams

None

8 SSD Mappings

The following SSDs are mapped to this use case:

S-1157: [*Objective / Priority 2*] The System shall update analyst displays with newly acquired waveform data within 1 minute of acquisition.

S-1284: [*Threshold*] The System shall provide the Analyst the capability to view Waveform QC Masks.

S-1296: [*Threshold*] The System shall store the processing time period(s) during which each Waveform QC Mask was applied to the underlying waveform data.

S-1297: [*Threshold*] The System shall store the Waveform QC Masks applied to the waveform data used for each waveform processing operation.

S-1298: [*Threshold*] The System shall store the channel masked by each Waveform QC Mask.

S-1299: [*Threshold*] The System shall store the identity of the user or processing stage creating each Waveform QC Mask.

S-1300: [*Threshold*] The System shall store the identity of the user or processing stage modifying each Waveform QC Mask.

S-1301: [*Threshold*] The System shall store the identity of the user or processing stage removing each Waveform QC Mask.

S-1302: [*Threshold*] The System shall store the time of each Waveform QC Mask creation.

S-1303: [*Threshold*] The System shall store the time of each Waveform QC Mask removal.

S-1304: [*Threshold*] The System shall store the time of each Waveform QC Mask modification.

S-1306: [*Threshold*] The System shall store the Analyst's rationale for creating a Waveform QC Mask.

S-1307: [*Threshold*] The System shall store the Analyst's rationale for modifying a Waveform QC Mask.

S-1308: [*Threshold*] The System shall store the Analyst's rationale for removing a Waveform QC Mask.

S-1372: [*Threshold*] The System shall provide the Analyst the capability to view continuous beams.

S-1386: [*Threshold*] The System shall store the beam definition parameters for all beams.

S-1393: [*Threshold*] The System shall store all derived channels related to one or more signal detections.

S-1394: [*null*] The System shall store derived waveform data with no related signal detections for the Operational Processing Time Period.

S-1421: [*Threshold*] The System shall store all signal detections.

- S-1438:** [*Threshold*] The System shall store time domain measurements.
- S-1450:** [*Threshold*] The System shall store polarization feature measurements.
- S-1465:** [*Threshold*] The System shall store frequency domain waveform measurements.
- S-1486:** [*Threshold*] The System shall store fk spectra measurements.
- S-1532:** [*Threshold*] The System shall provide the Analyst the capability to reject an event hypothesis.
- S-1542:** [*Threshold*] The System shall not automatically perform network signal association affecting signal detections that the Analyst is actively reviewing.
- S-1574:** [*Threshold*] The System shall provide the System User the capability to view station quality metrics.
- S-1576:** [*Threshold*] The System shall store the station quality metrics for all stations for each event hypothesis.
- S-1579:** [*Threshold*] The System shall compute an event hypothesis quality metric using the event hypothesis quality statistics for each event hypothesis formed on the System.
- S-1580:** [*Threshold*] The System shall recompute the event hypothesis quality metric for an event hypothesis when any of the event hypothesis quality statistics used to calculate the event hypothesis quality metric are updated.
- S-1586:** [*Threshold*] The System shall provide the Analyst the capability to view event hypothesis quality metrics.
- S-1588:** [*Threshold*] The System shall store the event quality metric for each event hypothesis.
- S-1616:** [*Threshold*] The System shall provide the Analyst the capability to designate the preferred event hypothesis for each event.
- S-1618:** [*Threshold*] The System shall store up to 300 unique event hypotheses for each event.
- S-1619:** [*Threshold*] The System shall store the confidence level of each computed event hypothesis location uncertainty bound.
- S-1620:** [*Threshold*] The System shall store the type (i.e., confidence, coverage, or k-weighted with the associated weights) of each location uncertainty bound.
- S-1621:** [*Threshold*] The System shall store modeling uncertainties for model based predictions of signal detection measurements.
- S-1622:** [*Threshold*] The System shall store uncertainties for observed signal detection measurements.
- S-1623:** [*Threshold*] The System shall store the sum squared weighted residual for each event hypothesis location.
- S-1624:** [*Threshold*] The System shall store the defining/non-defining state for each signal detection measurement associated to a stored event hypothesis.

- S-1625:** [*Threshold*] The System shall store a preferred event hypothesis for each event for each processing stage.
- S-1626:** [*Threshold*] The System shall store the processing stage during which each event hypothesis location was created.
- S-1627:** [*Threshold*] The System shall store the processing stage during which an event hypothesis is modified.
- S-1628:** [*Threshold*] The System shall store the processing stage that rejected an event.
- S-1644:** [*Threshold*] The System shall provide the Analyst the capability to manually align waveforms.
- S-1645:** [*Threshold*] The System shall provide the Analyst the capability to align waveforms based on travel time differences.
- S-1646:** [*Threshold*] The System shall provide the Analyst the capability to align waveforms based on optimal lag calculated by waveform cross correlation.
- S-1663:** [*Threshold*] The System shall store uncertainties for all event hypothesis magnitude estimates.
- S-1664:** [*Threshold*] The System shall store each single station magnitude estimate for each event hypothesis.
- S-1665:** [*Threshold*] The System shall store each network magnitude estimate for each event hypothesis.
- S-1666:** [*Threshold*] The System shall store the defining/non-defining state for each station magnitude associated to a stored event hypothesis.
- S-1711:** [*Threshold*] The System shall store the type of ground motion used by moment tensor calculations.
- S-1712:** [*Threshold*] The System shall store the filter applied to observed and synthetic waveforms when computing moment tensor solutions.
- S-1713:** [*Threshold*] The System shall store the Green functions used to compute a moment tensor solution.
- S-1714:** [*Threshold*] The System shall store the Earth models used to compute a moment tensor solution.
- S-1715:** [*Threshold*] The System shall store the elements of moment tensor solutions.
- S-1716:** [*Threshold*] The System shall store the percentage of deviatoric moment tensor solutions belonging to the double couple components.
- S-1717:** [*Threshold*] The System shall store the double couple fault plane solution computed from a moment tensor solution.
- S-1718:** [*Threshold*] The System shall store the scalar seismic moment computed from a moment tensor solution.
- S-1719:** [*Threshold*] The System shall store the station specific goodness of fit between theoretical and observed waveforms for moment tensor solutions.
- S-1735:** [*Threshold*] The System shall store the ϵ value computed for moment tensor solutions.

- S-1736:** [*Threshold*] The System shall store the k value computed for moment tensor solutions.
- S-1737:** [*Threshold*] The System shall store the uncertainty bounds on ϵ and k computed for moment tensor solutions.
- S-1738:** [*Threshold*] The System shall store the confidence level of uncertainty bounds on ϵ and k computed for moment tensor solutions.
- S-1816:** [*Threshold*] The System shall store the earth model and version used to compute an earth model prediction.
- S-1817:** [*Threshold*] The System shall store the corrections applied to earth model predictions.
- S-1818:** [*Threshold*] The System shall store the correction surface used to correct an earth model prediction.
- S-1819:** [*Threshold*] The System shall store the predicted slowness computed from a basemodel.
- S-1820:** [*Threshold*] The System shall store the uncertainties of a predicted slowness computed using a basemodel.
- S-1821:** [*Threshold*] The System shall store the predicted azimuths computed using a phase-specific basemodel.
- S-1822:** [*Threshold*] The System shall store the uncertainties of predicted azimuths computed using a basemodel.
- S-1823:** [*Threshold*] The System shall store the predicted travel-times computed from a basemodel.
- S-1824:** [*Threshold*] The System shall store the uncertainties of predicted travel-times computed using a basemodel.
- S-1842:** [*Threshold*] The System shall store predicted amplitude attenuation.
- S-1843:** [*Threshold*] The System shall store predicted amplitude attenuation uncertainties.
- S-1876:** [*Threshold*] The System shall notify Analysts working in a common processing stage if they are concurrently modifying event hypotheses for an event.
- S-1877:** [*Threshold*] The System shall notify Analysts working in a common processing stage if they are concurrently modifying signal detections in the same analysis time interval.
- S-1878:** [*Threshold*] The System shall provide the Analyst the capability to access and view all waveform data stored on the System.
- S-1885:** [*Threshold*] The System shall display 24 hours of continuous waveform data before the waveform displays flatline.
- S-1888:** [*Threshold*] The System shall provide the Analyst the capability to analyze continuous waveform data from within a selected time block.
- S-1892:** [*Threshold*] The System shall provide the Analyst the capability to iteratively undo/redo back to the last saved state of an event.

- S-1893:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on distance from event.
- S-1894:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on station name.
- S-1895:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on station probability of detection for an event.
- S-1896:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for a beam.
- S-1897:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for an array.
- S-1898:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for a 3-component station.
- S-1899:** [*Threshold*] The System shall provide the Analyst the capability to zoom the time axis of the waveform display.
- S-1900:** [*Threshold*] The System shall provide the Analyst the capability to scale the amplitude of individual displayed waveforms.
- S-1901:** [*Threshold*] The System shall provide the Analyst the capability to view a group of waveforms with their amplitudes scaled relative to the amplitudes of the other waveforms in the group.
- S-1902:** [*Threshold*] The System shall provide the Analyst the capability to adjust the height of an individual waveform display.
- S-1903:** [*Threshold*] The System shall provide the Analyst the capability to scroll waveform data along the time axis.
- S-1904:** [*Threshold*] The System shall provide the Analyst the capability to scroll the channels shown on the waveform display.
- S-1905:** [*Threshold*] The System shall provide the Analyst the capability to select the number of channels that are simultaneously visible on the waveform display.
- S-1906:** [*Threshold*] The System shall provide the Analyst the capability to time align waveforms based on observed signal detections for a user selected phase.
- S-1907:** [*Threshold*] The System shall provide the Analyst the capability to time align waveforms based on real time.
- S-1908:** [*Threshold*] The System shall provide the Analyst the capability to align waveforms based on theoretical signal detections for a user selected phase.
- S-1915:** [*Threshold*] The System shall provide the Analyst the capability to process data without altering another Analyst's existing solution.
- S-1917:** [*Threshold*] The System shall provide the Analyst the capability to mark an event as a reference event.

- S-1920:** [*Threshold*] The System shall provide the Analyst the capability to view any saved event hypothesis.
- S-1921:** [*Threshold*] The System shall provide the Analyst the capability to enter comments for an event hypothesis.
- S-1922:** [*Threshold*] The System shall provide the Analyst the capability to view comments for an event hypothesis.
- S-1927:** [*Threshold*] The System shall provide the Analyst the capability to select signal detections as processing input based on a time interval for an entire network during an analysis session.
- S-1928:** [*Threshold*] The System shall provide the Analyst the capability to select signal detections as processing input based on a time interval for a selected subset of stations during an analysis session.
- S-1929:** [*Threshold*] The System shall provide the Analyst the capability to individually select signal detections as processing input during an analysis session.
- S-1930:** [*Threshold*] The System shall provide the Analyst the capability to store new event hypotheses created during interactive processing.
- S-1946:** [*Threshold*] The System shall synchronize the System User's displays based on user actions.
- S-1947:** [*Threshold*] The System shall implement user interfaces according to the User Interface Guidelines.
- S-1959:** [*Threshold*] The System shall provide the System User the capability to view on-line help.
- S-1967:** [*Threshold*] The System shall store results from all stages of data processing.
- S-1985:** [*Threshold*] The System shall provide the Analyst the capability to view event hypothesis data on an interactive map.
- S-1986:** [*Threshold*] The System shall provide the Analyst the capability to view associated and unassociated signal detections on an interactive map.
- S-1987:** [*Threshold*] The System shall provide the Analyst the capability to view station data on an interactive map.
- S-1988:** [*Threshold*] The System shall provide the Analyst the capability to view geographic data on an interactive map.
- S-1996:** [*Threshold*] The System shall provide the Analyst the capability to access geospatial data.
- S-1997:** [*Threshold*] The System shall provide the Analyst the capability to save geospatial data.
- S-1998:** [*Threshold*] The System shall provide the Analyst the capability to determine the spatial relationships of geospatial data.
- S-1999:** [*null*] The System shall provide the Analyst the capability to view tabular listings of the results of spatial processing of geospatial data.
- S-2000:** [*null*] The System shall provide the System User the capability to view graphical images of the results of spatial processing of geospatial data.

- S-2042:** [*Threshold*] The System shall store automatic and interactive processing parameters in the database.
- S-2043:** [*Threshold*] The System shall store automatic and interactive processing results in the database for use by subsequent processing.
- S-2044:** [*Threshold*] The System shall store the relation of processing results to processing parameters in the database.
- S-2164:** [*Threshold*] The System shall access requested waveform data within one (1) minute of receipt by the Data Processing Partition.
- S-2166:** [*Threshold*] The System shall automatically process late-arriving waveform data within one (1) minute of receipt by the Data Processing Partition.
- S-2167:** [*Threshold*] The System shall write a 6 hour or less time block of 40Hz waveform data within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)
- S-2168:** [*Threshold*] The System shall read a 6 hour or less time block of 40Hz waveform data outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)
- S-2169:** [*Threshold*] The System shall read a 6 hour or less time block of 40Hz waveform data within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)
- S-2170:** [*Threshold*] The System shall write a 6 hour or less time block of 40Hz waveform data from outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)
- S-2223:** [*Threshold*] The System shall store all data and derived processing results to non-volatile memory / persistent storage as soon as the data and/or derived processing results are available.
- S-2417:** [*null*] The System shall store hydroacoustic signal detection groups
- S-2420:** [*Threshold*] The System shall provide the Analyst the capability to store selected derived waveforms.
- S-2588:** [*Threshold*] The System shall set the processing stage workflow status of events to reflect analysis activity.
- S-2603:** [*Threshold*] The System shall provide the System User the capability to access requested waveform data.
- S-2604:** [*Threshold*] The System shall provide the Analyst the capability to access late-arriving waveform data within one (1) minute of receipt by the Data Processing Partition.
- S-3025:** [*null*] The System shall provide the Analyst the capability to create a signal detection template from an existing event.

9 Notes

1. In this UCR and all of its child UCRs (e.g. "Refines Event Location", "Refines Event Magnitude", etc.), the display classes store computed results to transient storage via the OSD mechanism as the event is refined in order to trigger execution of configured processing sequences. These processing sequences are configured by

the System Maintainer (see "Defines Processing Sequence" UCR), and are automatically executed by the Processing Sequence Control mechanism in response to OSD callbacks (the Processing Sequence Control mechanism is shown in "System Detects Events" UCR). These changes to the Event Hypothesis are stored with private visibility such that the changes are accessible only to the Analyst work session where the changes are being made; other Analysts cannot see the updates until the Analyst chooses to save the Event Hypothesis, at which time the Event Hypothesis is stored to persistent storage via the OSD mechanism. The storage of an Event Hypothesis to persistent storage may trigger additional processing sequences, as defined by the System Maintainer.

2. The Analyst may undo/redo editing operations while refining the event, but only back to the last save.
3. See "Marks Processing Stage Complete" UCR for a state machine diagram for Event Completion Status.
4. This UCR covers creation of signal detection templates. The Analyst applies such templates when building new events (see "Builds Event" and "Scans Waveforms and Unassociated Detections" UCRs).

10 Open Issues

1. Instead of just setting a flag on the Event to mark it as a reference event, consider an "event catalog" concept instead (see CR 1085).
2. Need to clarify the scope of conflict checking w.r.t. intervals and processing stages (CR 1567)

11 Change History

1. E1 Iteration Review (3/14) - Fully Described
 - a. Initial release
2. E2 Review (10/2014) - Delivered
 - a. Got rid of "event reservation" concept (just warn instead). Simplified modeling of warnings.
 - b. Renamed "COI" mechanism to "Object Storage and Distribution"
 - c. Changes to better support conflict checking.
 - d. OSD now supports data lifespan (transient vs. persistent) in addition to visibility (private vs. global).
 - e. Added ability for Analyst to refresh display.
 - f. Added ability for Analyst to mark an event as a reference event (to account for SSD-2108 and SSD-2109)
 - g. Analyzes Events Display -> Refines Event Display aggregation now has multiplicity of 0..1 (previously was '*').
 - h. Added Event Completion Status class and got rid of Event Analysis Status class.
 - i. Added note about using predicted travel times to limit subscription timeframe for waveforms and detections.
 - j. Added support for creating Signal Detection Template.
 - k. Added diagram showing subdisplays.
3. E3 Update (12/11/14)
 - a. Signal Detection Template and Event Hypothesis now inherit from Processing Result class (CR 1447)
 - b. Event must be saved when creating a Signal Detection Template (CR 1447)
 - c. Conflict checker now iterates over detections instead of events (CR 1447)
 - d. Added support for copying an event (CR 1211)
 - e. Got rid of "best hypothesis" field on Event class, added expansion flow (CR 1446)
 - f. Pass EH to use as starting point into Open Event (CR 1446)
 - g. Now show use of Event History Display (CR 1532)
 - h. Added mention of term "active review" to Event class and Add Active Analyst() method (CR 1442)
 - i. Got rid of "Scan Status" concept (CR 1449)

j. Changed most alternate flows to expansion flows (per new modeling convention).