SAND2015-4474R

Use Case Realization Report
UCR-02.08 System Refines Event Location

generated: Apr 13, 2015 12:39 PM

1 Use Case Description

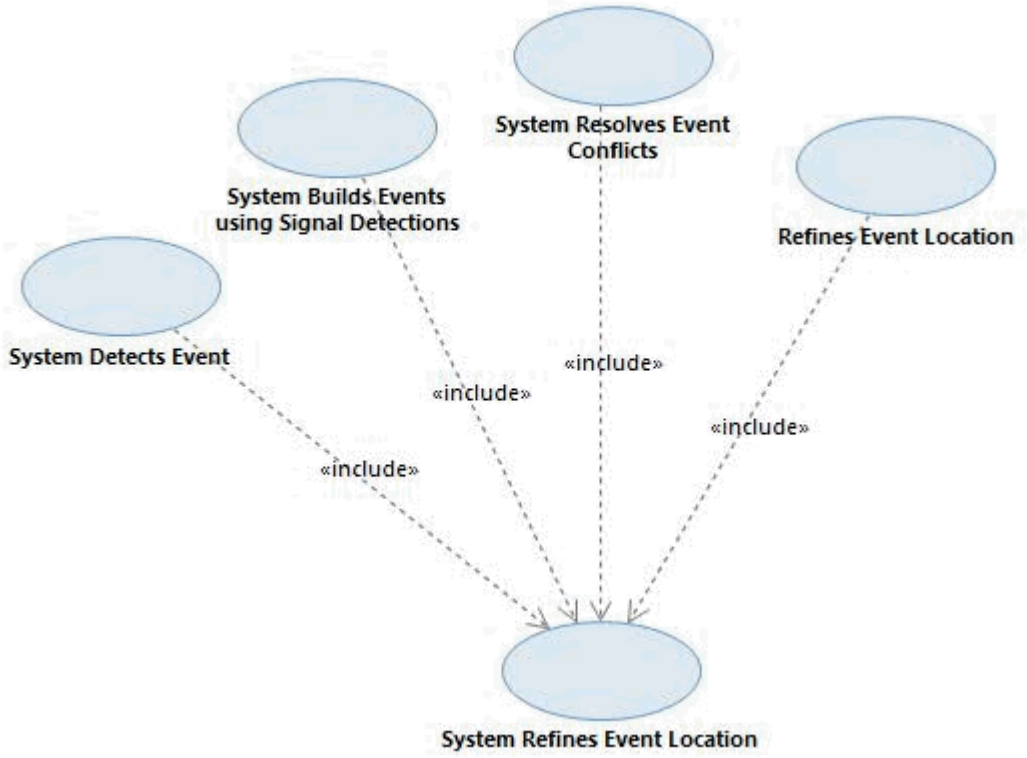
This architecturally significant use case describes how the System refines event hypothesis location solutions. The System locates events by finding the event location minimizing the difference between signal detection feature measurements and signal detection feature predictions (see 'System Measures Signal Features' UC). The System references both empirical knowledge from past events and geophysical models to form the signal detection feature predictions. The System also computes an uncertainty bound for each event hypothesis location solution describing a region bounding the event hypothesis' hypocenter and origin time at a particular confidence level. The System creates a variety of location solutions for each event hypothesis. These location solutions vary from one another in either the input parameters the System uses or in the location solution components the System restrains to fixed values (e.g. depth) during event location calculations. The System computes location solutions using input parameters configured by the System Maintainer (see 'Configures Processing Components' UC). The Analyst has the option to override input parameters originally configured by the System Maintainer (see 'Refines Event Location' UC).

This use case is architecturally significant due to the processing and memory resource consumption of 3D earth model calculations.

2 Architecture Description

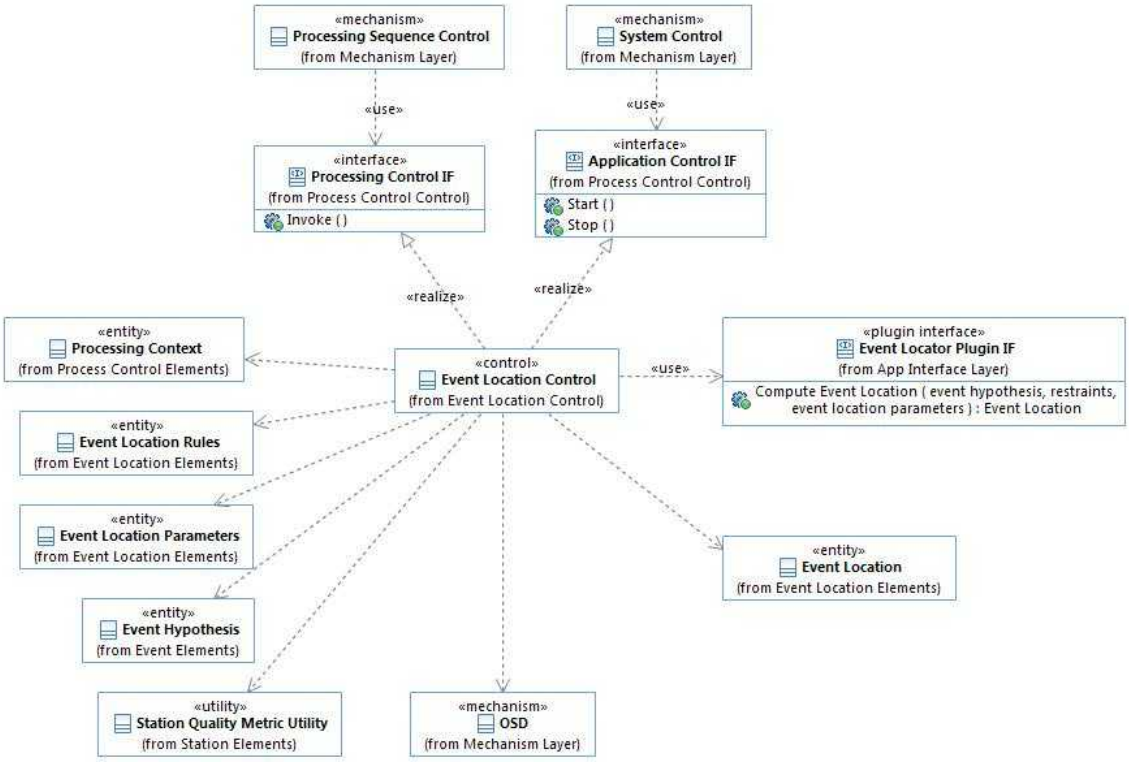
The Event Location Control class is responsible for controlling event location computations. Event Location Control may be invoked by Processing Sequence Control as part of executing a step in a Processing Sequence (see "System Detects Event" UCR), or manually invoked by an Analyst as part of refining an event (see "Refines Event Location" UCR). Event Location Control uses an Event Locator to perform the event location calculations. Multiple Event Locator plugins exist in the system, each realizing a common plugin interface. The specific Event Locator plugin used varies dynamically at runtime based on the Event Location Parameters. When invoked from Processing Sequence Control, Event Locator Control builds up the Event Location Parameters to be passed to the Event Locator, selects and invokes the appropriate Event Locator plugin based on those parameters, updates the Event Hypothesis with the results, and stores the Event Hypothesis via the OSD mechanism. Note that the stored Event Hypothesis is only accessible to OSD clients executing within compatible Processing Contexts (e.g. changes made within an Analyst work session may only be accessible within that session until the Analyst saves the event hypothesis).

3 Use Case Diagram



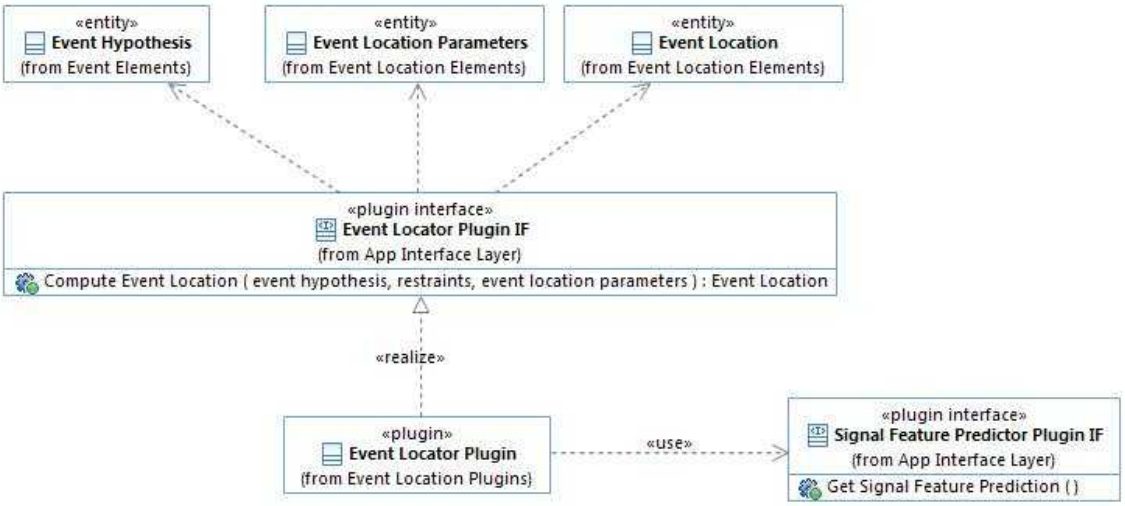
4 Class Diagrams

4.1 Classes - Event Location Control



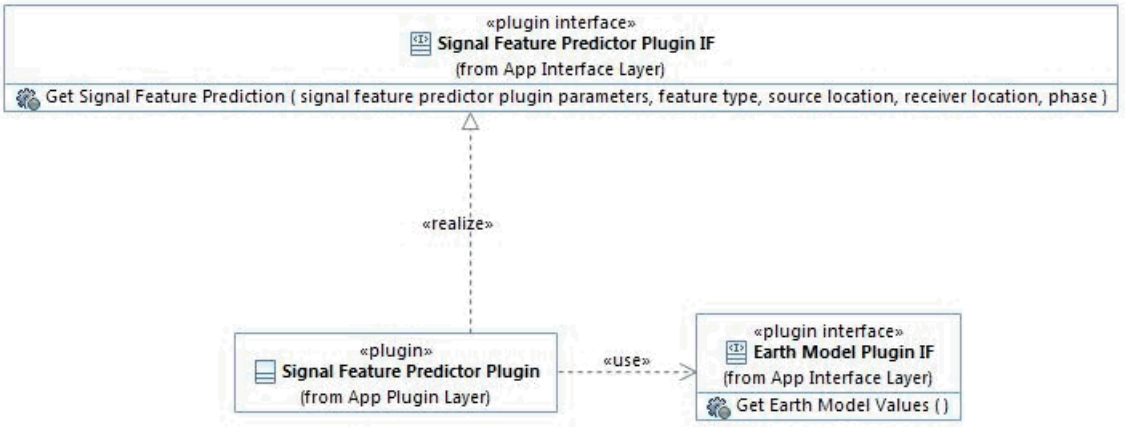
This diagram shows the Event Location Control class and related classes.

4.2 Classes - Event Locator Plugin IF



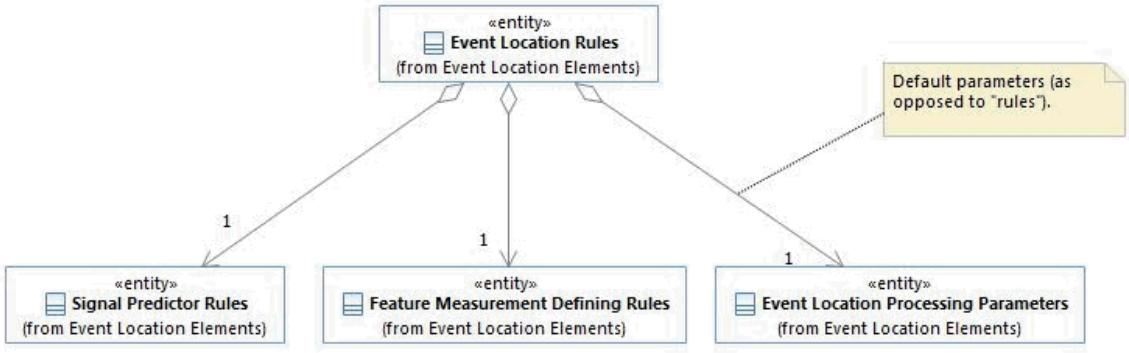
This diagram shows the Event Locator Plugin IF interface, which defines the common interface that all event locator plugins must realize. Also shown is a Residual Based Event Locator Plugin, which is just one possible instance of an event locator plugin. In this case, the Residual Based Event Locator Plugin requires a Signal Feature Predictor Plugin, since feature predictions are needed to compute residuals. Other event locator plugins may not need the Signal Feature Predictor Plugin, and/or may need to use different plugins.

4.3 Classes - Signal Feature Predictor Plugin IF



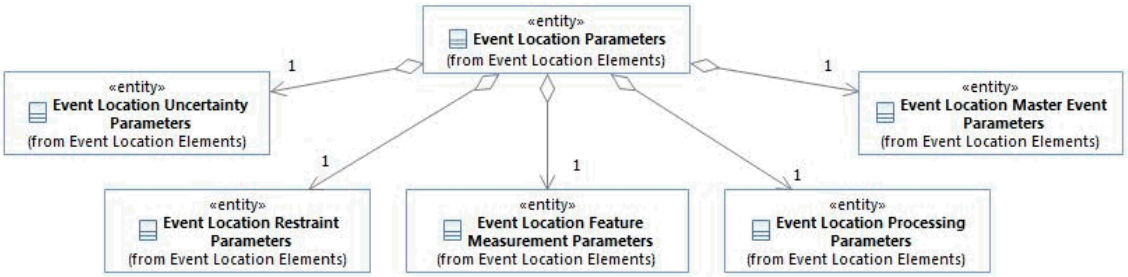
This diagram shows the Signal Feature Predictor plugin and related elements. Note that the "Signal Feature Predictor" class shown above is a conceptual class that represents any/all of the possible signal feature predictors on the system. As such, the dependency to Earth Model IF above may not apply for some specific predictors.

4.4 Classes - Event Location Rules



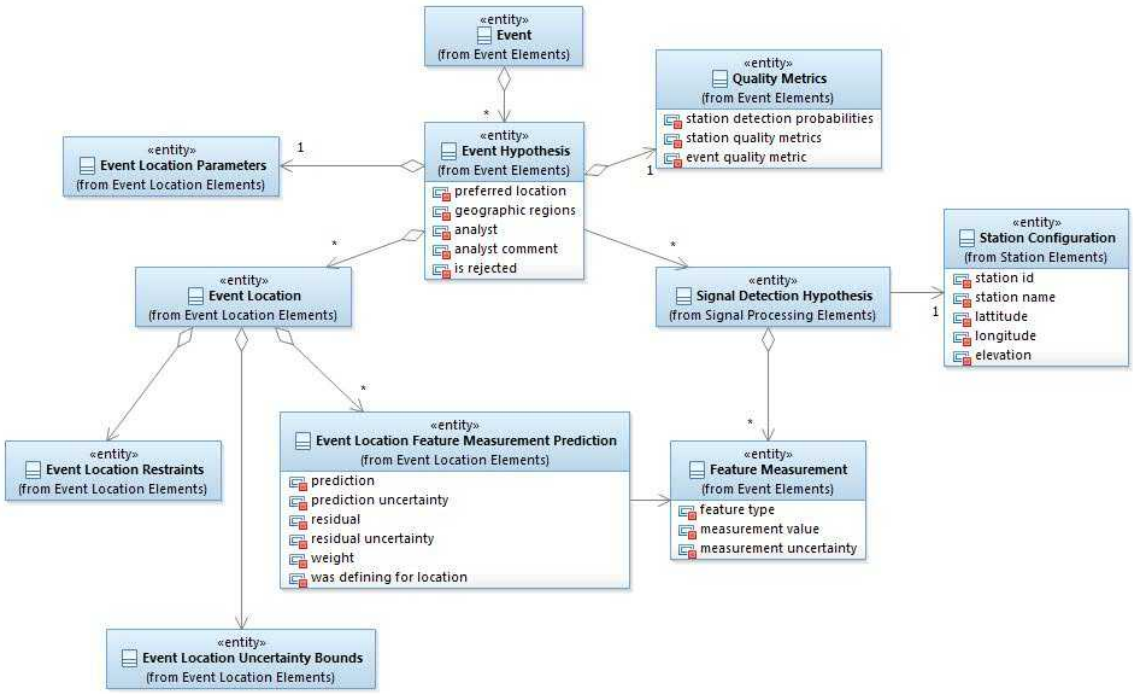
This diagram shows details of the Event Location Rules class. Event Location Rules are configured by the System Maintainer (see "Configure Processing Components" UCR), and are used by Event Locator Control to build up the Event Location Parameters that are provided as input to the Event Locator.

4.5 Classes - Event Location Parameters



This diagram shows details of the Event Location Parameters class. The Event Location Parameters class is provided as input the Event Locator to control the algorithm behavior.

4.6 Classes - Event Hypothesis



This diagram shows the portions of an Event Hypothesis that are used to compute event location as well as the portions that are computed by the algorithm. The event location algorithm analyzes feature measurements of

signal detections for one or more event hypotheses. The algorithm uses the station location associated with each signal detection as well as features of the detection to compute the Event Location, which is the primary output of the event location algorithm. Multiple different Event Locations may be computed for each Event Hypothesis, each with a different set of Event Location Restraints. The location algorithm also computes an Event Location Feature Measurement Prediction for each analyzed feature measurement. The Event Location Parameters class captures the parameters that were provided as input to the event location algorithm, enabling subsequent Analysts to recompute the location with the same parameters.

5 Class Descriptions

<<control>> Event Location Control
Responsible for controlling the event location computation. Retrieves necessary data, invokes the appropriate Event Locator to compute the new location, and stores the result.

<<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>> Event Location
Represents a computed location for an event.

<<entity>> Event Location Feature Measurement Parameters
Represents event location parameters for specific feature measurements. The parameters include the following for each feature measurement provided to the event locator:

- Whether the feature measurement should be defining for event location
- Whether the algorithm is free to toggle the defining state
- Which signal prediction model to use
- Which signal predictor to use

<<entity>> Event Location Feature Measurement Prediction
Represents the information about a Feature Measurement prediction that is computed by the event location algorithm. The residual uncertainty is computed from measurement uncertainty and prediction uncertainty.

<<entity>> Event Location Master Event Parameters
Represents event location parameters pertaining to locating an event based on a specified "master event" (i.e. Master Event Relocation). Includes the following info from the master event:

- The ID of the master event
- The location and location uncertainty of the master event
- For each signal detection on the master event that also exists on the event under refinement (i.e. signal detections with the same channel and phase), includes related feature measurements and associated uncertainties pertinent to location computation from the master event

<<entity>> Event Location Parameters

Represents the parameters that are input to the Event Locator. Initially computed by the system based on the Event Location Rules defined by the System Maintainer, but may be overridden by the Analyst when refining events.

<<entity>> Event Location Processing Parameters
General settings for controlling the behavior of the event location computation (e.g. settings for event location algorithm, max number of iterations).

<<entity>> Event Location Restraint Parameters
Represents restraints on the location coordinate spaces (lat, lon, depth or time) for the event location computation. Restraints indicate which coordinate spaces are restrained and the associated restrained value (or value range) to be used for that coordinate.

<<entity>> Event Location Restraints
Represents the restrained coordinate values that were used during computation of the associated event location.

<<entity>> Event Location Rules
Rules used for initial refinement of event locations. These rules are configured by the System Maintainer.

<<entity>> Event Location Uncertainty Bounds
Represents the uncertainty bounds for the associated event location.

<<entity>> Event Location Uncertainty Parameters
Represents the type of uncertainty bound (Confidence, Coverage or K-Weighted), confidence level and scaling factor for the locator to use when computing event location uncertainty.

<<entity>> Feature Measurement
Represents the value and uncertainty of a measured feature of a signal detection.

<<entity>> Feature Measurement Defining Rules
Rules that specify which types of Feature Measurement(s) can be defining for event location for different geographic regions, stations, channels and/or phases.

<<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>> Quality Metrics
Represents quality metrics for a single event hypothesis. This includes the event quality metric, station quality metrics at the time the event occurred, and station probabilities of detecting the event.

<<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 Predictor Rules
Rules that specify which signal predictors and earth models to use for different geographic regions, stations, channels and/or phases.

<<entity>> Station Configuration
Represents the configurable aspects of a station such as its name, location, whether it is enabled, etc.

<<interface>> Application Control IF
Defines the interface implemented by all <<control>> classes in the system that are controlled by System Control.

<<interface>> Processing Control IF
Defines the interface implemented by all <<control>> classes in the system that are controlled by the Processing Sequence Control <<mechanism>>.

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

<<mechanism>> Processing Sequence Control
Mechanism for executing and controlling processing sequences configured by the System Maintainer.

<<plugin interface>> Earth Model Plugin IF
Standard interface for all Earth Model plugins. All Earth Model plugins in the system realize this interface.

<<plugin interface>> Event Locator Plugin IF
Standard interface for all Event Locator plugins. All Event Locator plugins in the system realize this interface. Note that the Event Location Master Relocation Parameters portion of the Event Location Parameters object is optional, and is only applicable when requesting a Master Event Relocation. In addition, some Event Locator plugin implementations may not support Master Event Relocation.

<<plugin interface>> Signal Feature Predictor Plugin IF
Standard interface for all Signal Feature Predictor plugins. All Signal Feature Predictor plugins in the system realize this interface.

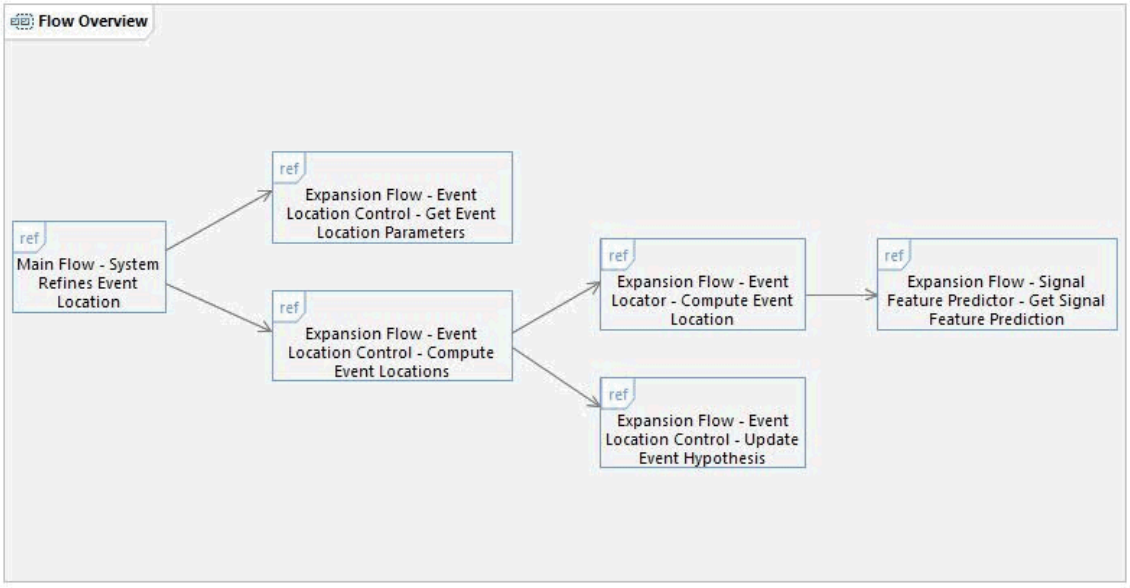
<<plugin>> Event Locator Plugin
Abstract class that represents any/all of the Event Locators that may be plugged in to the system behind the Event Locator IF plugin interface. Event Locators are responsible for calculating event locations.

<<plugin>> Signal Feature Predictor Plugin
Abstract class that represents any/all of the Signal Feature Predictors that may be plugged in to the system behind the Signal Feature Predictor IF plugin interface. Signal Feature Predictors are responsible for calculating signal feature predictions.

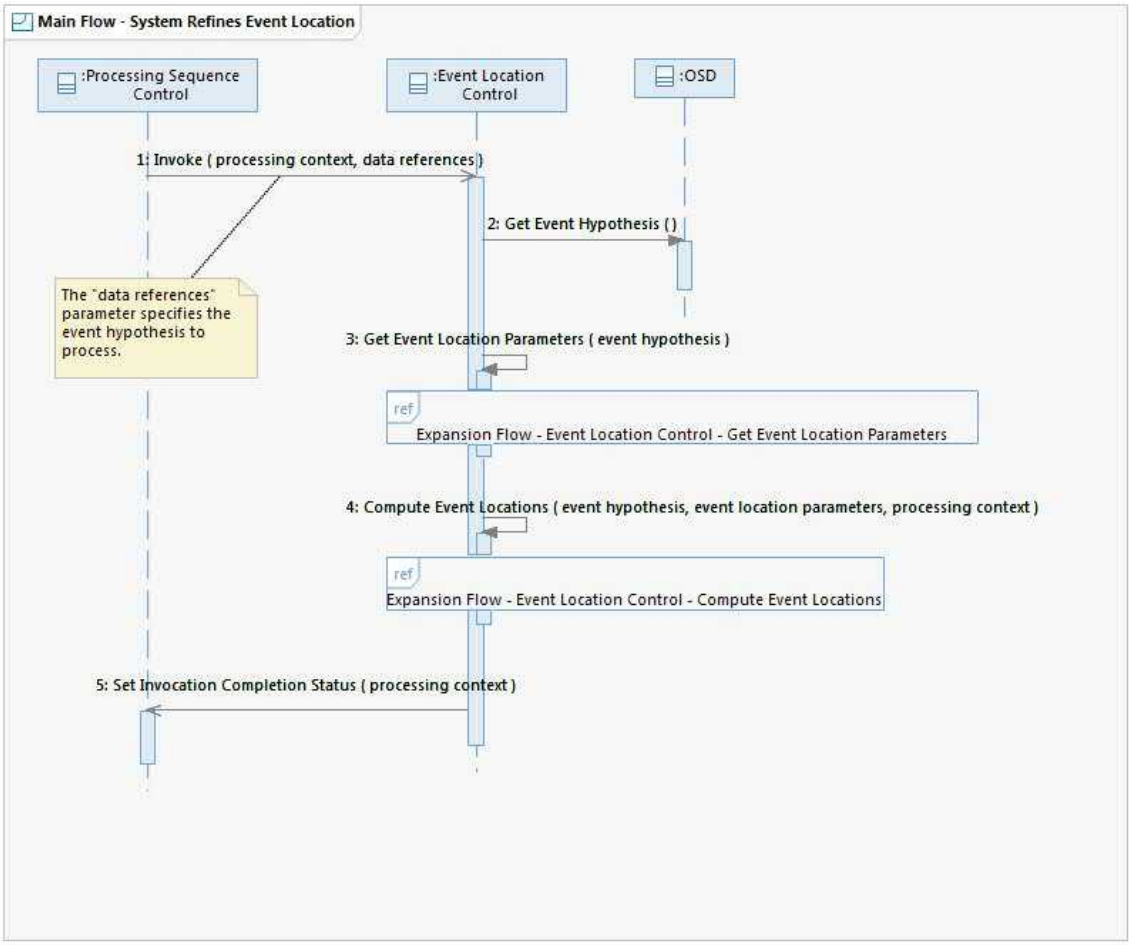
<<utility>> Station Quality Metric Utility
Utility class that computes the station quality metric.

6 Sequence Diagrams

6.1 Flow Overview



6.2 Main Flow - System Refines Event Location



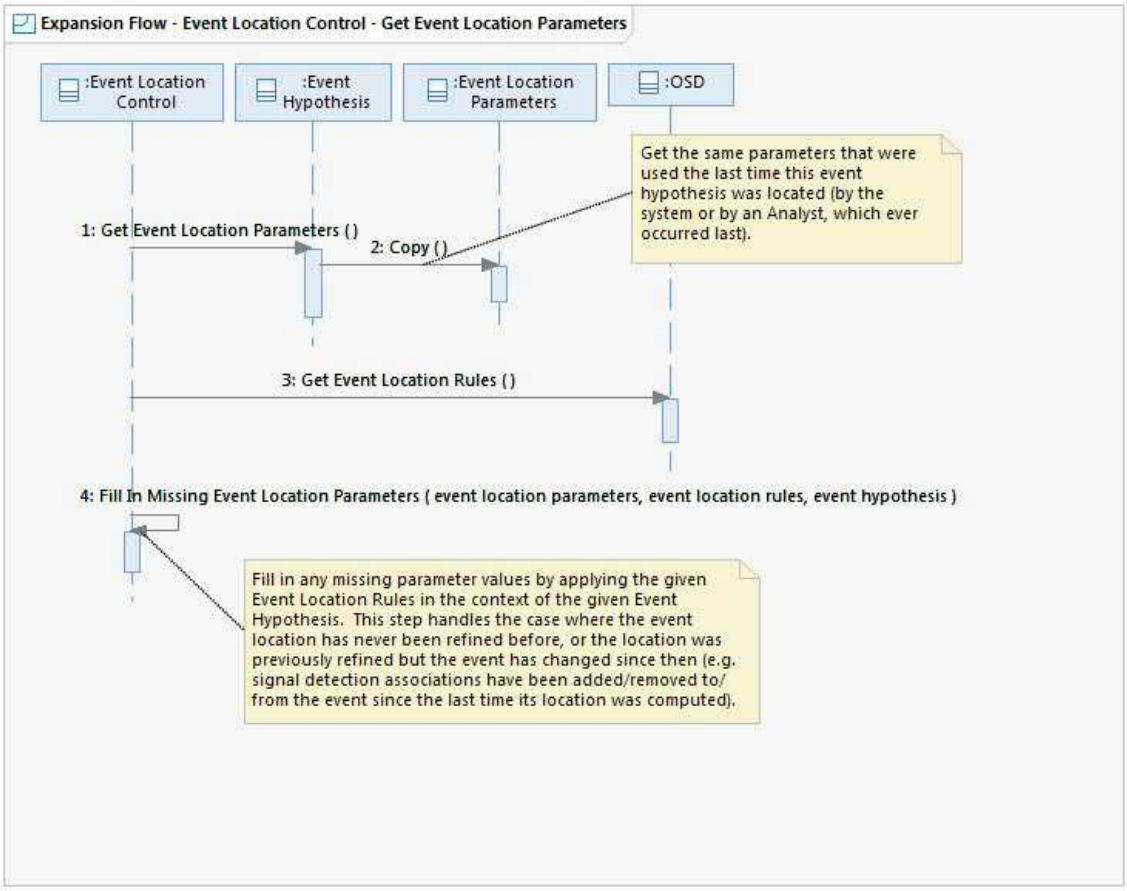
This flow shows how the system refines event location. This flow is stimulated by the Processing Sequence Control mechanism as part of executing an automatic processing sequence. The precise triggering conditions for such sequences are configured by the System Maintainer (see "Defines Processing Sequence" UCR). For

more information about the Processing Sequence Control mechanism see "System Detects Event" UCR.

6.2.1 Operation Descriptions

None

6.3 Expansion Flow - Event Location Control - Get Event Location Parameters

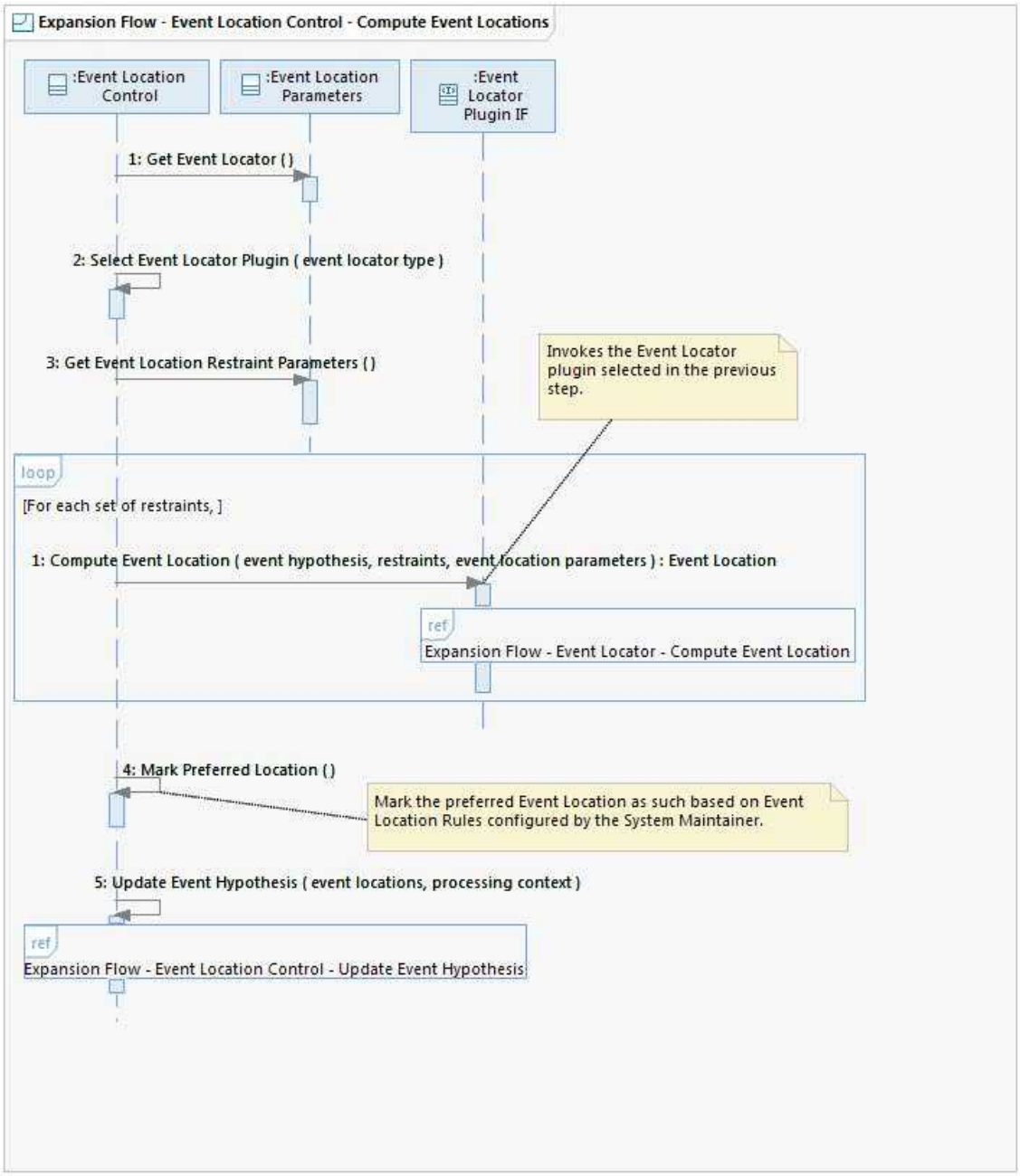


This flow shows how the Event Location Control class builds up the Event Location Parameters object to pass to the Event Locator. Note that this flow may also be invoked directly from the Refines Event Location Display (see "Refines Event Location" UCR).

6.3.1 Operation Descriptions

None

6.4 Expansion Flow - Event Location Control - Compute Event Locations

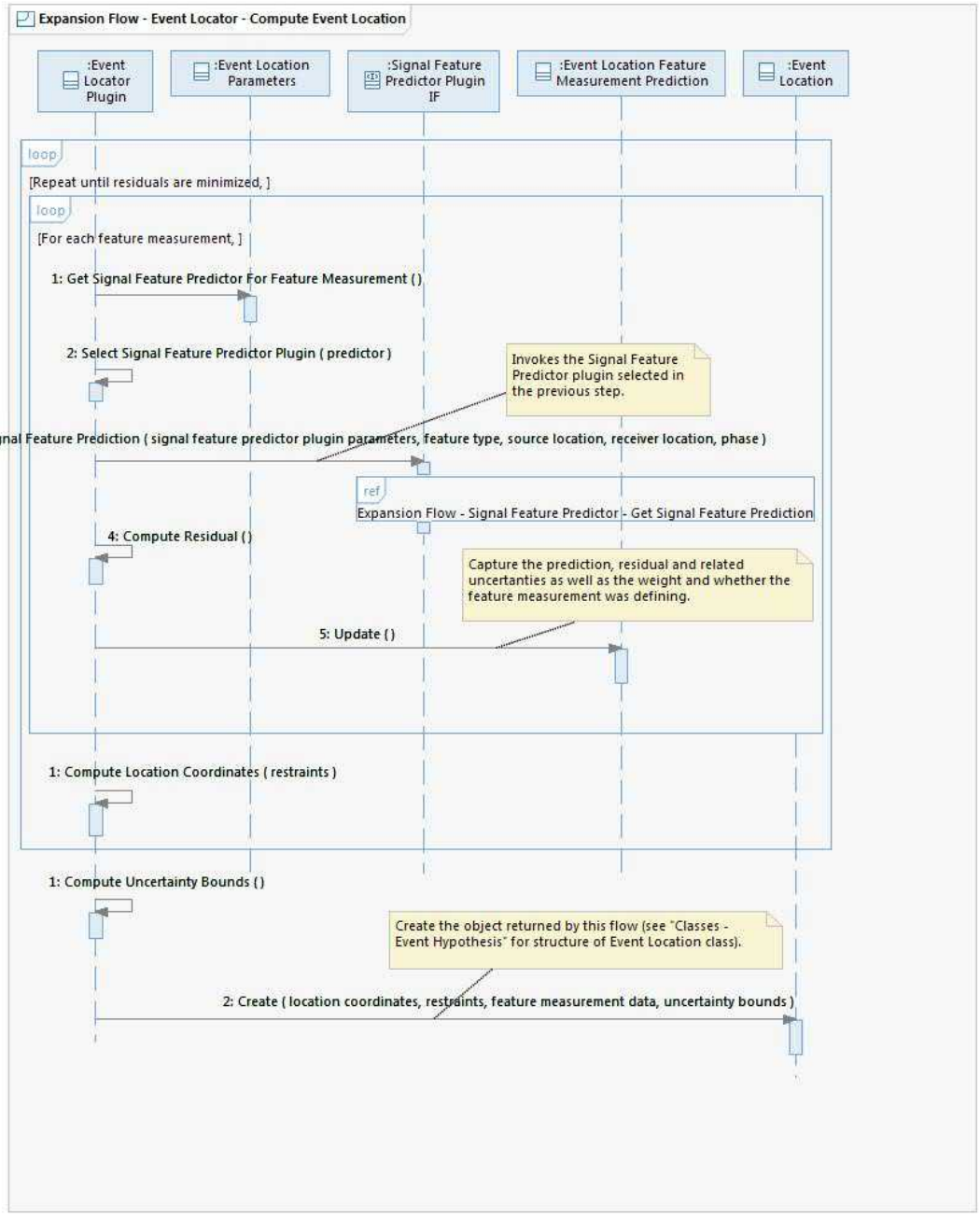


This flow shows how the Event Location Control invokes the Event Locator to compute locations for an event hypothesis. The control class selects which Event Locator to invoke based on the event locator type specified in the Event Location Parameters, then invokes the Event Locator through the Event Locator IF plugin interface. The Event Locator returns Event Location Results. Event Location Control updates the Event Hypothesis with the Event Location Results. Note that this flow may also be invoked directly from the Refines Event Location Display (see "Refines Event Location" UCR).

6.4.1 Operation Descriptions

Operation: Event Location Control::Select Event Locator Plugin()
Select and bind to the specific event locator plugin that corresponds to the given event location algorithm.

6.5 Expansion Flow - Event Locator - Compute Event Location



This flow notionally shows how a particular Event Locator plugin might compute location for an event hypothesis. The flow shown here may not apply to all Event Locator plugins. In this example, the Event Locator plugin iteratively computes residuals between observed vs. predicted feature measurements and the resulting location coordinates until the residuals no longer improve. To compute predicted feature measurements, the Event Locator uses a Signal Feature Predictor plugin. The specific predictor used may vary for each prediction based on parameters and/or rules specified in the Event Location Parameters class.

As a possible variation of this flow for performing Master Event Location, the flow might look essentially the same except that instead of using a Signal Feature Predictor to get the predicted feature measurements the Event Locator might use the feature measurements associated with a designated "master event" (obtained from the Event Location Master Event Parameters class).

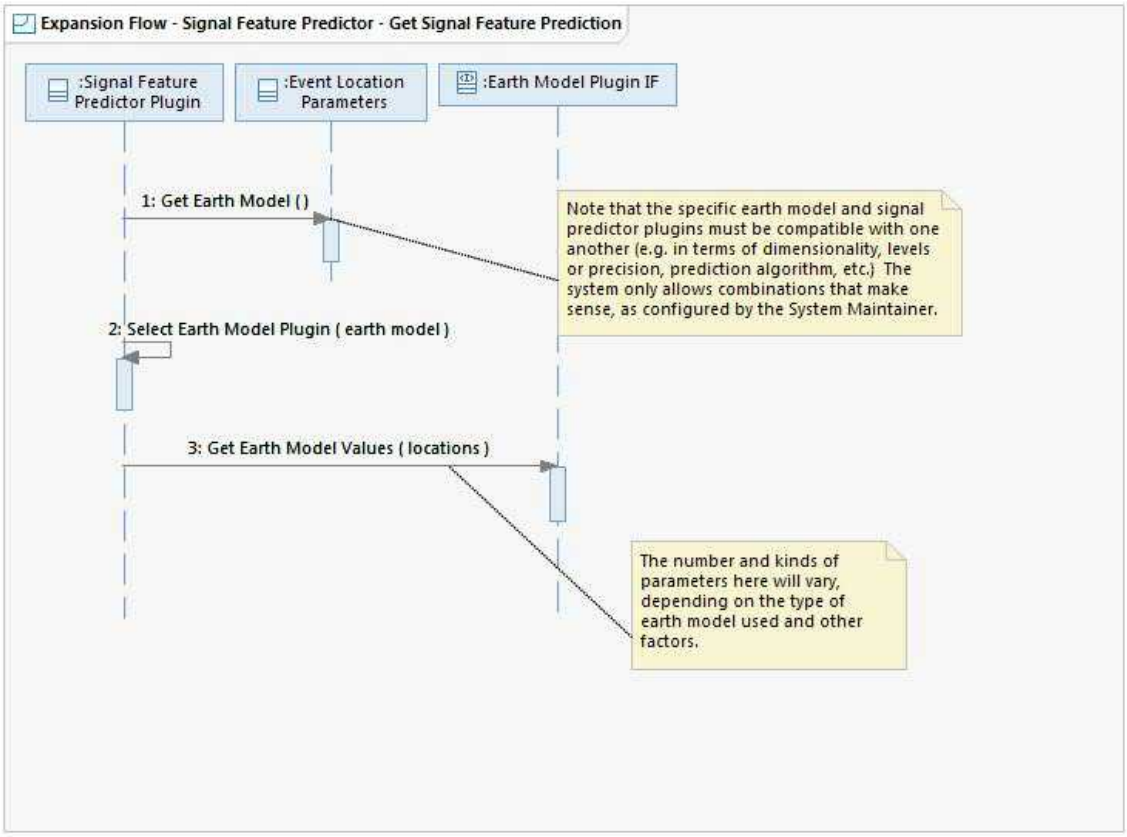
6.5.1 Operation Descriptions

Operation: Event Locator Plugin::Select Signal Feature Predictor Plugin()

Select and bind to the specific Signal Feature Predictor plugin that corresponds to the given signal feature predictor.

Operation: Event Locator Plugin::Compute Residual()
Compute the difference between the signal feature measurement and the signal feature prediction, as well as the residual uncertainty.

6.6 Expansion Flow - Signal Feature Predictor - Get Signal Feature Prediction

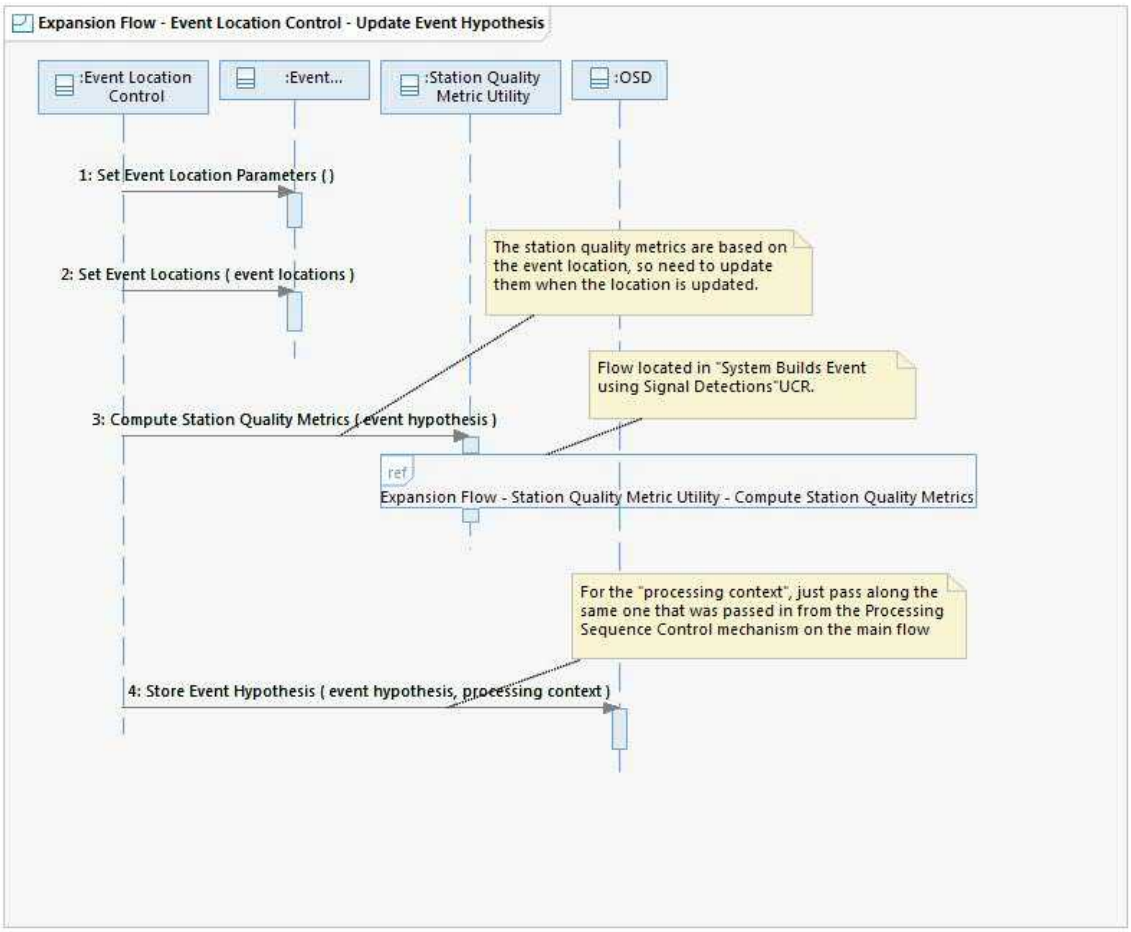


This flow notionally shows how a particular Signal Feature Predictor plugin might compute predicted feature measurements. The flow shown here may not apply to all Signal Feature Predictor plugins. In this example, the Signal Feature Predictor uses an Earth Model plugin to obtain model values needed for the prediction. The specific earth model used is dependent on parameters and/or rules specified in the Event Location Parameters class.

6.6.1 Operation Descriptions

None

6.7 Expansion Flow - Event Location Control - Update Event Hypothesis



This flow shows how the Event Location Control class updates the Event Hypotheses passed in from Processing Sequence Control with the Event Location objects returned by the Event Locator and the Event Location Parameters previously computed in "Expansion Flow - Event Location Control - Get Event Location Parameters". Event Location Control also recomputes the station quality metrics based on the updated event location.

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.

7 State Machine Diagrams

None

8 SSD Mappings

The following SSDs are mapped to this use case:

S-1563: [*Threshold*] The System shall locate event hypotheses found using waveform correlation processing using the same location algorithms as events found using other types of event processing.

S-1564: [*Threshold*] The System shall calculate signal detection feature measurement uncertainties for signal detections found using waveform correlation using the cross correlation coefficient.

S-1572: [*Threshold*] The System shall compute the station quality metric for all stations for each event hypothesis.

S-1576: [*Threshold*] The System shall store the station quality metrics for all stations for each event hypothesis.

S-1592: [*Threshold*] The System shall compute event hypothesis relocations using any combination of the seismic, hydroacoustic, and infrasound event hypothesis relocation parameters.

S-1593: [*Threshold*] The System shall compute event hypothesis relocations using any combination of the event hypothesis relocation parameters from a single station.

S-1594: [*Threshold*] The System shall compute event hypothesis relocations using any combination of the event hypothesis relocation parameters from multiple stations.

S-1595: [*Threshold*] The System shall compute event hypothesis relocation uncertainty bounds from event hypothesis location covariance matrices and event hypothesis location uncertainty bound scaling factors.

S-1596: [*Threshold*] The System shall compute the uncertainty coverage ellipse for each event hypothesis relocation.

S-1600: [*Threshold*] The System shall set the defining/non-defining state for signal detection measurements during event hypothesis relocation processing.

S-1601: [*Threshold*] The System shall compute modeling uncertainties for model based predictions of signal detection measurements.

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-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-1631: [*Threshold*] The System shall compute event hypothesis relocations using teleseismic and regional seismic signal detections.

S-1640: [*Threshold*] The System shall perform master event relocation using travel time differences.

S-1653: [*Threshold*] The System shall compute new event hypothesis magnitude estimates when a new event hypothesis location is computed.

S-1776: [*Threshold*] The System shall use correction surfaces to compute corrections to earth model predictions.

S-1777: [*Threshold*] The System shall apply earth model prediction corrections to earth model predictions computed from basemodels.

S-1779: [*Extensibility*] The System shall compute predicted slowness using a one-dimensional phase-specific basemodel.

S-1780: [*Threshold*] The System shall compute phase-specific slowness predictions using a velocity model where the velocity of the Earth varies as a function of depth but not latitude or longitude.

S-1781: [*Extensibility*] The System shall compute the uncertainties of predicted slowness computed using a one-dimensional phase-specific basemodel.

S-1782: [*Threshold*] The System shall compute the uncertainty of phase-specific slowness predictions using a velocity model where the velocity of the Earth varies as a function of depth but not latitude or longitude.

S-1783: [*Extensibility*] The System shall compute predicted slowness using a three-dimensional phase-specific basemodel.

S-1784: [*Threshold*] The System shall compute phase-specific slowness predictions using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

S-1785: [*Extensibility*] The System shall compute the uncertainties of predicted slowness computed using a three-dimensional phase-specific basemodel.

S-1786: [*Threshold*] The System shall compute the uncertainty of phase-specific slowness predictions using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

S-1787: [*Extensibility*] The System shall compute predicted azimuths using a three-dimensional phase-specific basemodel.

S-1788: [*Threshold*] The System shall compute phase-specific azimuth predictions using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

S-1789: [*Extensibility*] The System shall compute the uncertainties of predicted azimuths computed using a three-dimensional phase-specific basemodel.

S-1790: [*Threshold*] The System shall compute uncertainty of phase-specific azimuth predictions using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

S-1791: [*Extensibility*] The System shall compute predicted travel-times using a one-dimensional phase-specific basemodel.

S-1792: [*Threshold*] The System shall compute phase-specific travel-time predictions using a velocity model where the velocity of the Earth varies as a function of depth but not latitude or longitude.

S-1793: [*Extensibility*] The System shall compute the uncertainties of predicted travel-times computed using a one-dimensional phase-specific basemodel.

S-1794: [*Threshold*] The System shall compute the uncertainty of phase-specific travel-time predictions using a velocity model where the velocity of the Earth varies as a function of depth but not latitude or longitude.

S-1795: [*Extensibility*] The System shall compute predicted travel-times using a two-dimensional phase-specific basemodel.

S-1796: [*Threshold*] The System shall compute phase-specific travel-time predictions using a velocity model where the velocity of the Earth varies as a function of latitude and longitude but not depth.

S-1797: [*Threshold*] The System shall compute predicted travel time of Rayleigh waves and Love waves using frequency-specific group and phase velocity models where the group/phase velocity varies as a function of latitude and longitude but not depth.

S-1798: [*Extensibility*] The System shall compute the uncertainties of predicted travel-times computed using a two-dimensional phase-specific basemodel.

S-1799: [*Threshold*] The System shall compute phase-specific uncertainty of predicted travel-time using a velocity model where the velocity of the Earth varies as a function of latitude and longitude but not depth.

S-1800: [*Threshold*] The System shall compute uncertainty of predicted travel time of Rayleigh waves and Love waves using frequency-specific group and phase velocity models where the group/phase velocity varies as a function of latitude and longitude but not depth.

S-1801: [*Extensibility*] The System shall compute predicted travel-times using a three-dimensional phase-specific basemodel.

S-1802: [*Objective / Priority 1*] The System shall compute phase-specific travel-time predictions using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

S-1803: [*Extensibility*] The System shall compute the uncertainties of predicted travel-times computed using a three-dimensional phase-specific basemodel.

S-1804: [*Extensibility*] The System shall compute phase-specific uncertainty of predicted travel-time using a velocity model where the velocity of the Earth varies as a function of latitude, longitude, and depth.

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-1852: [*Objective / Priority 1*] The System shall incorporate monthly variations in travel time for

hydroacoustic data.

S-1853: [*Objective / Priority 1*] The System shall use a seasonal climatological model for computing travel times in infrasound data.

S-1855: [*Objective / Priority 1*] The System shall model thermospheric phases when computing infrasonic travel times.

S-1856: [*Objective / Priority 1*] The System shall model Lamb waves when computing travel times in infrasound data.

9 Notes

1. Although Event Location Master Event Parameters are passed to all Event Locator plugins (as part of the Event Location Parameters), some Event Locator plugin implementations may not support Master Event Relocation.

2. Multiple Event Relocation (such as Joint Hypocenter Determination) is considered to be a specialized research activity and is thus covered in "Analyzes Research Event" UCR instead of this use case.

3. This UCR shows the system refining the location for one event hypothesis at a time. However, in the actual implementation it may be desirable to support relocating a batch of events at a time, for efficiency. Batching is not shown in this UCR since it is considered a design/implementation optimization.

4. Computation of the event quality metric after an event is relocated is configurable (see 'Defines Processing Sequence' UC). Configuration could include selection of a minimum change in location that would result in recomputing the event quality metric or selection of when to recompute the event quality metric based on the cause for the event relocation.

10 Open Issues

1. Need to update how "configuration", "rules" and "parameters" are modeled to be consistent with other UCRs. (CR 1522)

2. Consider refactoring the location-related classes into a more logical structure (CR 1560).

11 Change History

1. E1 Review (03/2014) - Fully Described
   a. Initial release

2. E2 Review (10/2014) - Delivered
   a. Event Locator Control now invokes Event Locator Plugin multiple times (once for each set of restraints).
   b. On flow diagram, now show calculation of location, residuals, etc. more explicitly.
   c. Renamed "COI" mechanism to OSD
   d. OSD now supports data lifespan (transient vs. persistent) in addition to visibility (private vs. global).
   e. Added support for Master Event Relocation.

3. E3 Update
   a. Recompute station quality metrics after relocating event (CR 1453).