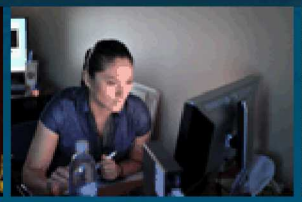
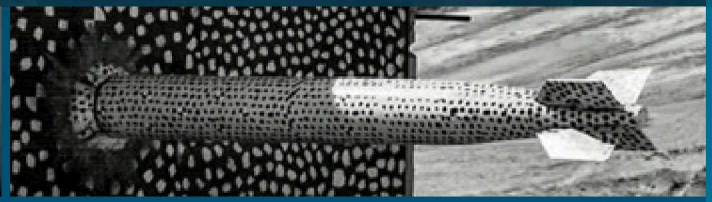
SAND2020-3357PE

Geophysical Monitoring System (GMS)
Global Associator (GA)





PRESENTED BY

Stephen Heck

3/24/2020

SAND2020-????

GMS Global Associator Requirements

Specifications (SSD)

- Section 3.2.6.1 Network Signal Association
- Section 3.2.6.2 Late Network Signal Association
- Section 3.2.6.3 Waveform Correlation
- Section 3.2.6.4 Conflict Resolution
- Section 3.2.6.5 Station Quality Metric
- Section 3.2.6.6 Event Hypothesis Quality Metric

Use Cases

- Section 2.6 System Builds Events using Signal Detections

User Interface Storyboards (UIS)

- N/A

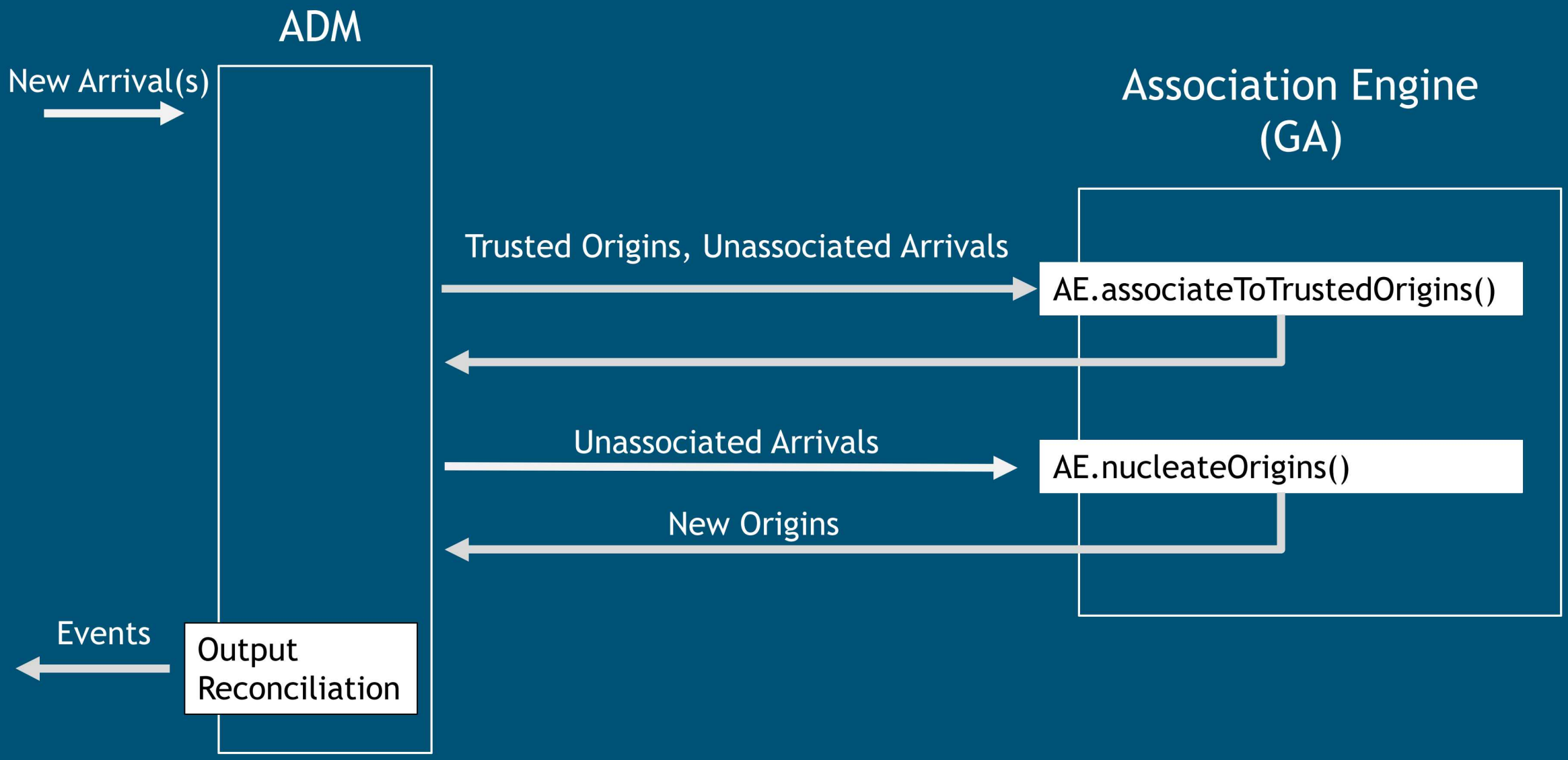
Current USNDC Implementation

- C code implementation
- Processing flows through a series of heuristic modules/steps
- Modules have been added over time
- Rebuilds events at each time step
- Fixed time-window processing
- GA philosophy: Build all possible event hypotheses and gradually filter down to the "best" events.

Global Associator Prototyping Motivation

- Data-driven Association with Association Data Management library (ADM)
  - Separate associator state management from the specifics of the association algorithm
- Build a working knowledge of the GA algorithm
  - Extensive testing on synthetic and real data
- Opportunity to review GA design/implementation from the GMS design perspective
  - Easy to maintain/extend implementation
  - Modern distributed processing frameworks
- Prototyping the association algorithm allows our team to iterate and adapt in a fashion that doesn't fit into the GMS development process.
- Give the GMS team an implementation that "bridges the gap" between requirements and the production system code
  - Develop new implementation in Java

Global Associator – Association Data Management (ADM)



Global Associator Prototyping Plan

- Start from scratch with Java
  - Reference IDC Processing Document
  - Reference C source code
- Start with minimal implementation of GA that will allow us to test ADM-GA
  - Grid search -> location -> conflict resolution
- Develop GeoTess GA models
- Test by gradually increasing difficulty
  - Start with synthetic events/arrivals
  - Test on real data
  - Use Event Commonality Score to measure similarity between reference and automated bulletins
- Test-driven development*
  - Constrain implementation to be testable
  - Unit-tests are the best documentation for what the code is supposed to accomplish

Global Associator Prototyping Accomplishments

ADM

- Implemented (7/2019)

GA implementation

- Conflict Resolution (7/2019)
- Station models using GeoTess (8/2019)
- Grid Search (8/2019)
- Chi-Squared Test (9/2019)
- Location Module (7/2019)

Synthetic Test Generation

- Implemented (12/2019)

ADM-GA Testing

- Perfect data (11/2019)
- Off-grid origins + perfect arrivals (11/2019)
- Off-grid origins + perturbed arrivals (01/2020)

Global Associator Prototyping Upcoming Work

Testing

- Overlapping events/arrivals (In progress – April 2020)
- Noise arrivals (April-May 2020)
- Real data testing May 2010 dataset (July 2020)
  - Reference bulletin (UGEB): www.sandia.gov/uueb
- Waveform correlation events (July 2020)

GA Components

- Large event processing (July/August 2020)
- Predicted phase search (July August 2020)

Global Associator Prototyping Summary

- We are developing a Java data-driven GA"-like" algorithm.
  - Grid-search -> chi-squared-> redundant event filter -> location -> conflict resolution
  - GA station model building code. Models are implemented in GeoTess

- Testing
  - Developed test library for generating synthetic tests
  - Gradually increasing test difficulty: Perfect data (easy) --- Real data (hard)
  - Real data testing with 2010 May dataset (IMS).