**Macintosh HD:Users:Dependable:Shared:Documents:Templates:DCi.logo.pdf**

Sponsor: AFRL

Project No.: 144

Contract: ASTRA IST

This document is for authorized use only. Any information contained herein should be considered proprietary unless otherwise specified.

© 2023 Dependable Computing. All Rights Reserved

Dependable Computing

1/10/2024

Top-Level Specification

Zone Alert Service

Abstract

This document provides the intended functionality of an OpenUxAS Zone Alert Service as a top-level solution specification. The specification is meant to only be as formal as is required to be reasonably accurate for all readers. The specification defines what functions/behaviors the solution provides and the model of how it provides then fitting within the OpenUxAS environment.

Changes

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Change** |
| V0.1 | 1/10/2024 | Initial draft |
| V0.2 | 1/22/2024 | Updates based on behavior validation |
| V0.3 | 1/24/2024 | Added issue tracking and issues for required OpenUxAS integration research |
|  |  |  |
|  |  |  |
|  |  |  |

.

Contents

1 Introduction 1

1. Introduction

The Zone Alert Service is a stand-up service that can be activated as part of an OpenUxAS instance. Its purpose is to alert subscribers to when vehicles associated with the UxAS instance in one of two states:

1. Are in current violation of a zone boundary.
2. Are in danger of violating a zone boundary based on current linear velocity and a lookahead time window.

These three cases are more carefully defined below.

1. Issues
2. Investigate where collision radius **RV** may be obtained for vehicles confirm it is defined in a vehicle configuration message for each vehicle, or whether another configuration variable is required either as part of vehicle state or as configuration parameter of the Zone Alert Service as a universal for each vehicle.
3. =Investigate the message needed to know what vehicles there are
4. Investigate the messages to know what keep in and keep out zones to compute against
5. Investigate the messages to know about what
6. Primary Functions

## Function 1: Report Path-Aware, Potential Unintended Zone Violations

### Definitions

1. **Zone Boundary Violation Event**: A *zone boundary violation* at time t is defined as one or more point in a circle of radius **RV** centered on vehicle position **VP** in the horizontal plane that intersects the boundaries and/or interiors of one or more *keep-out zones* and/or outside one or *more keep-in zones*.
2. **Potential future zone violation**: A vehicle is in potential future violation with a specific zone at time **t** if the vehicle’s **linear trajectory at time t** is a future **zone violation event** at one or more times from **t** to a future lookahead time **t**+**TW**, where **TW** is a defined constant of the running Zone Alert Service.
3. **Same potential future violation**: For a given vehicle and zone, two potential future zone violations at times **t1** and **t2** are the same future potential zone violation if the same potential future zone violation exists for all times from **t1** to **t2** based on the actual path of the vehicle from time **t1** to **t2**.
4. **Current filed path**: For a given vehicle, a current filed path is a declared mission path to be followed by the vehicle that has not yet been completed. If a filed vehicle point has been completed.
5. **Iterative Service**: An OpenUxAS service that computes and delivers its functional behavior at fixed, repeated soft real-time (best effort) intervals.
6. **Function Iteration**: An iteration of an OpenUxAS iterative service.

### Assumptions:

1. The zone alert service is aware of any pre-defined path that the vehicle is currently following.
2. Pre-defined paths are defined so as not to violate defined keep-in and keep-out zones for the path’s vehicle.
3. A ‘completed’ filed path is easy to discern from an uncompleted filed path.

### Function Definition:

1. At the iterative rate of the service, subscribers shall be notified of potential future zone violations satisfying the rules of this function.
2. The successive iterative period of the service is never greater than **TIt**.
3. Subscribers are notified of a potential future violation for each vehicle. For each vehicle **V**, where the last state report was **VS** at time **TS**, the current iteration time is at time **TP**, and the time window for detecting future violations is defined over **t** as **TS < TP <= t <= TS**, the following cases define behavior of detection:
4. **TP >TS+TW**: No violation reports can be produced for the vehicle **V** as position reports are too stale for the time window.
5. **TP <= TS+TW  and VS distance to Path Segment > ED**: Subscribers shall be notified of any **potential future violations** for projected violation that begins at time **TX** such that **TP < TX <= TS + TW.**
6. **TP <= TS+TW  and VS distance to Path Segment <= ED**: Subscribers shall not be notified of any **potential future violations** during this function iteration.

### Note:

This means if stale reports are provided for vehicle positions, vehicles that have drifted off course and into projected violations will not report them The less stale a vehicle state report, the more accurate the service. Timely state reports are required for this function to work well (availability issue).

## Function 2: Report Zone Boundary Violations

### Definitions:

* **Violated Zone**: A keep-in or keep-out zone at time t that is subject of one or more zone boundary violation events.

### Assumptions:

1. Vehicle position at time t is the point-position estimated for a vehicle at time t in OpenUxAS.
2. A vehicle position at time t may be an exact reported position from OpenUxAS or an estimated position based on a physics-based interpolation from last reported vehicle state.
3. Vehicle positions are exact, without noise, in the intended simulation environment.
4. There are a limited number of zones and vehicles, so that the number of simultaneous notifications to be sent and received will never burden OpenUxAS or services.

### Function Definition:

1. Subscribers shall be informed of current zone violation events given the last vehicle state report, where current is defined by the circle at the last reported vehicle position **VP** and collision radius **RV**.
2. A notification is sent for each vehicle for each zone violated.
3. Secondary Functions

The following functions will support the primary functions.

## Function 3: Initialize Service

### Definitions:

### Assumptions:

### Function Definition:

1. The service is initialized with values for the parameters The service accepts subscriptions during the intilization / setup phase of OpenUxAS.
2. The service subscribes to the messages required to monitor system state for the service:
   1. Entity Configurations
   2. Entity States
   3. Keep-In/Out Zones
   4. Operating Regions
   5. Routes or maybe UniqueAutomationResponse TBD
   6. AirVehicleState

### Example Cases:

## Function 4: Subscribe to Reports

### Definitions:

1. Interest subscription. A party subscribes to interest in a mission group, as defined by the message profile, during OpenUxAS initiation phase.

### Assumptions:

1. We can create a message for entities that want to subscribe to zone alerts.
2. For the current use case, Amase entities, Controller, and other services might be interested in this subscription.

### Function Definition:

1. The service accepts subscriptions during the initialization / setup phase of OpenUxAS

### Example Cases:

## Function A: Internal State Awareness

### Definitions:

### Assumptions:

1. The service receives and processes all relevant messages during initialization, planning, and execution phases of OpenUxAS

### Function Definition:

1. During the initialization phase of OpenUxAS, the service listens for and processes relevant declarations:
   1. The service listens to all entity definitions from AMASE and catalogues the entities
   2. The service listens to operating region declarations and catalogues them
   3. The service listens to keep-in and keep-out zone declarations and catalogies them
2. During operation the service listens for and processes all relevant state updates:
   1. Waypoint plans from UniqueAutomationResponse messages during planning phase
   2. AirVehicleState messages from AMASE during execution phase.

### Example Cases: