Hazard Analysis
Software Engineering

Team 17, Track a Trace
Zabrain Ali
Linqi Jiang
Jasper Leung
Mike Li
Mengtong Shi
Hongzhao Tan

October 19, 2022

Table 1: Revision History

Date	**Developer(s)**	**Change**
Oct. 19, 2022	Zabrain Ali	Modified Hazard Analysis
Oct. 19, 2022	Jasper Leung	Modified Hazard Analysis
Oct. 19, 2022	Hongzhao Tan	Modified Hazard Analysis
Oct. 19, 2022	Mengtong Shi	Modified Hazard Analysis
Oct. 19, 2022	Mike Li	Modified Hazard Analysis
Oct. 19, 2022	Linqi Jiang	Modified Hazard Analysis

Contents

List of Tables

1 Introduction

This document is the hazard analysis for the PyERT system. PyERT is a software toolkit that aims to reverse engineering the GERT toolkit. PyERT is intended to re-implement the features in GERT that use ArcGIS Pro packages with open-source packages and libraries to make it fully open-source and independent from proprietary software like ArcGIS Pro. The definition of a hazard is a condition or property in the system together with a condition in the environment that has the potential to cause harm or damage.

2 Scope and Purpose of Hazard Analysis

The scope of the document is to identify potential hazards within the system components, the steps to mitigate the hazards, and the resulting safety and security requirements.

3 System Boundaries and Components

The system that is referred to in this document that the hazard analysis will be conducted on consists of:

1. The script of the PyERT toolkit, including following 4 major components:
   - GPS Points Preprocessing
   - Classification of GPS Points Segments
   - Alternative Routes
   - Activity Locations Information
2. The physical computer of the user

The system boundary hence includes the script of the toolkit which is the implementation of the toolkit the users will run on their devices to achieve their goals. The physical computer is not controlled by the PyERT toolkit but by the user, and is still an important element of the system for hazard analysis since it is the environment the script will be executed in.

4 Critical Assumptions

- The project assumes the users will directly run the compiled .pyc files of the source code .py files with through the command line to use the product.

5 Failure Mode and Effect Analysis

The following pages contain a failure modes and effect analysis (FMEA) table of the PyERT system.

Table 2: Failure Mode and Effect Analysis

Component	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref.
GPS Points Preprocessing	1. Missing data 2. Missing input	1. Inaccurate GPS point 2. Missing GPS points	1a. Faulty data 2a. Incorrect data format	1a. Replace missing data with other values such as mean, median, mode, random sampling, interpolation. Delete any data that is detected to be faulty. 2a. Return an error message stating that the format of the user's inputted data was not correct	1a. IR2 2a. IR3	1. HR1-1 2. HR1-2
Classification of GPS Points Segments	1. Incorrect travel mode	1. Inaccuracy of data for specific travel modes	1a. Error with the Multinomial Logit Model used by the Mode Detection Module that is not successfully extracting GPS data into different travel modes	1a. Classify unknown segments by adjacent/similar segments	1a. IR4	1. HR2-1
Generating Routes and Alternatives	1. Route cannot be generated from the given data	1a. Unable to generate output containing routes 1b. Unable to use route for analysis	1a. Faulty data 1b. Given GPS points don't form a route	1a. Same as HR1-1-1a 1b. Return error output to user stating a route could not be generated from the given data	1a. IR2 1b. IR5	1. HR3-1

Continued on next page

Table 2 Continued from previous page

Component	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref.
Activity Locations Information	1. No stop or end points of trip segment	1. Unable to generate potential activity locations	1a. Faulty data 1b. No significant period of time detected where the location of the GPS point doesn't change	1a. Same as HR1-1-1a 1b. Modify conditions needed to determine potential activity locations.	1a. IR2 1b. IR6	1. HR4-1
General	1. Program closes unexpectedly	1a. Current process is lost 1b. No output can be generated	1a. Instability in program or the user's system causes crash 1b. The user's system loses power	1a. Reopen the program and check the error generated by the program 1b. Same as H5-1-1a	1a. IR7	1. HR5-1

Concluded

6 Safety and Security Requirements

Using the results from the FMEA, we can add the following safety and security requirements to our already existing safety and security requirements specified in the Software Requirements Specification. New requirements will be highlighted in **bold**.

6.1 Safety Requirements

N/A

6.2 Security Requirements

6.2.1 Privacy Requirements

PR1. **The program shall not store a user's personal information.**

- Rationale: To ensure that a user's information and privacy is maintained.
- Associated Hazards: *N/A*

6.2.2 Audit Requirements

AR1. **All revisions to the program shall be visible on GitHub.**

- Rationale: To allow audits for all versions of the program.
- Associated Hazards: *N/A*

6.2.3 Integrity Requirements

IR1. **The program shall not use any files other than the ones the user has provided and the ones that are included with the program and Python.**

- Rationale: To ensure the user's information and privacy is maintained, no external files should be accessed.
- Associated Hazards: *N/A*

IR2. **The program shall attempt to correct the user's inputted data if data is detected to be missing or faulty.**

- Rationale: An error should be generated even if there are small errors in the user's input, to ensure ease of use for the user.
- Associated Hazards: HR1-1, HR3-1, HR4-1

IR3. **The program shall return an error message if the file containing the data is in the wrong format.**

- Rationale: The program should only be designed to accept csv inputs of a certain format, so unexpected program behaviour is avoided. The user should be informed of this format error.
- Associated Hazards: HR1-2

IR4. **Segments with unknown detected travel modes shall be re-processed and compared to similar segments so the travel modes can be determined.**

- Rationale: The data returned to the user should be as complete as possible, so the program should attempt to classify as many travel modes as possible.
- Associated Hazards: HR2-1

IR5. **If the given GPS points cannot form a route, an error shall be returned to the user.**

- Rationale: The user should know if the data they used cannot be used to form routes.
- Associated Hazards: HR3-1

IR6. **If potential activity locations cannot be determined, the conditions for finding potential activity should be modified to find potential activity locations.**

- Rationale: The data returned to the user should be as complete as possible,so the program should attempt determine as many activity locations as possible.
- Associated Hazards: HR4-1

IR7. **If the program closes unexpectedly, an error should be returned the next time the user opens the program.**

- Rationale: The user should be informed if the program unexpectedly crashes.
- Associated Hazards: HR5-1

7 Roadmap

The hazard analysis resulted in many new security requirements being added to the already existing requirements from the Software Requirements Specification. Due to time constraints, not all of the security requirements will be implemented. AR1, IR1, IR3, IR5 and IR7 will be implemented before the end of the capstone, while the other requirements may not be implemented before the end of the capstone.