Automatically Testing a Conversational Crowd Computing Agent

Orestis Kanaris - Supervised by: Sihang Qiu, Ujwal Gadiraju, Jie Yang - CSE3000

1. Introduction:

To ensure the smooth running of a Conversational Crowd Computing Platform, by testing its robustness and responsiveness.

2. Motivation:

- Lack of available research on automatically testing conversational crowd computing platforms.
- Ensure the robustness and responsiveness of Dandelion.

Definitions:

Robustness: Robustness, as defined in IEEE standard 24765 : 2010, is the degree to which a system operates correctly under exceptional inputs or stressful environmental conditions [2].
Responsiveness: Ensuring response time to be under 30 seconds [3].

3. Research Questions:

1. How to automatically test the robustness of a conversational crowd computing platform.
2. How to reliably measure the responsiveness of a chatbot for crowd computing.

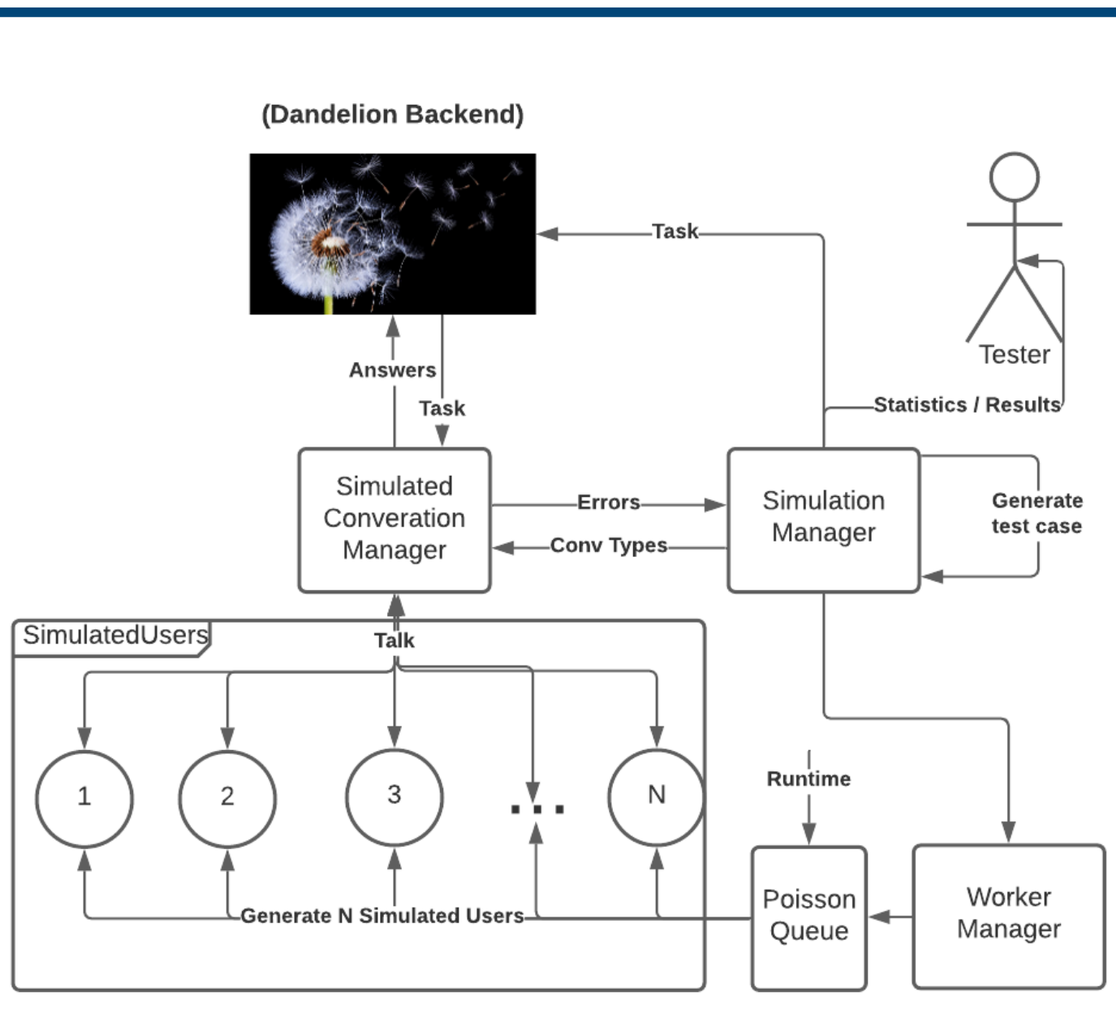

Figure 1: UML of the MAS Simulation

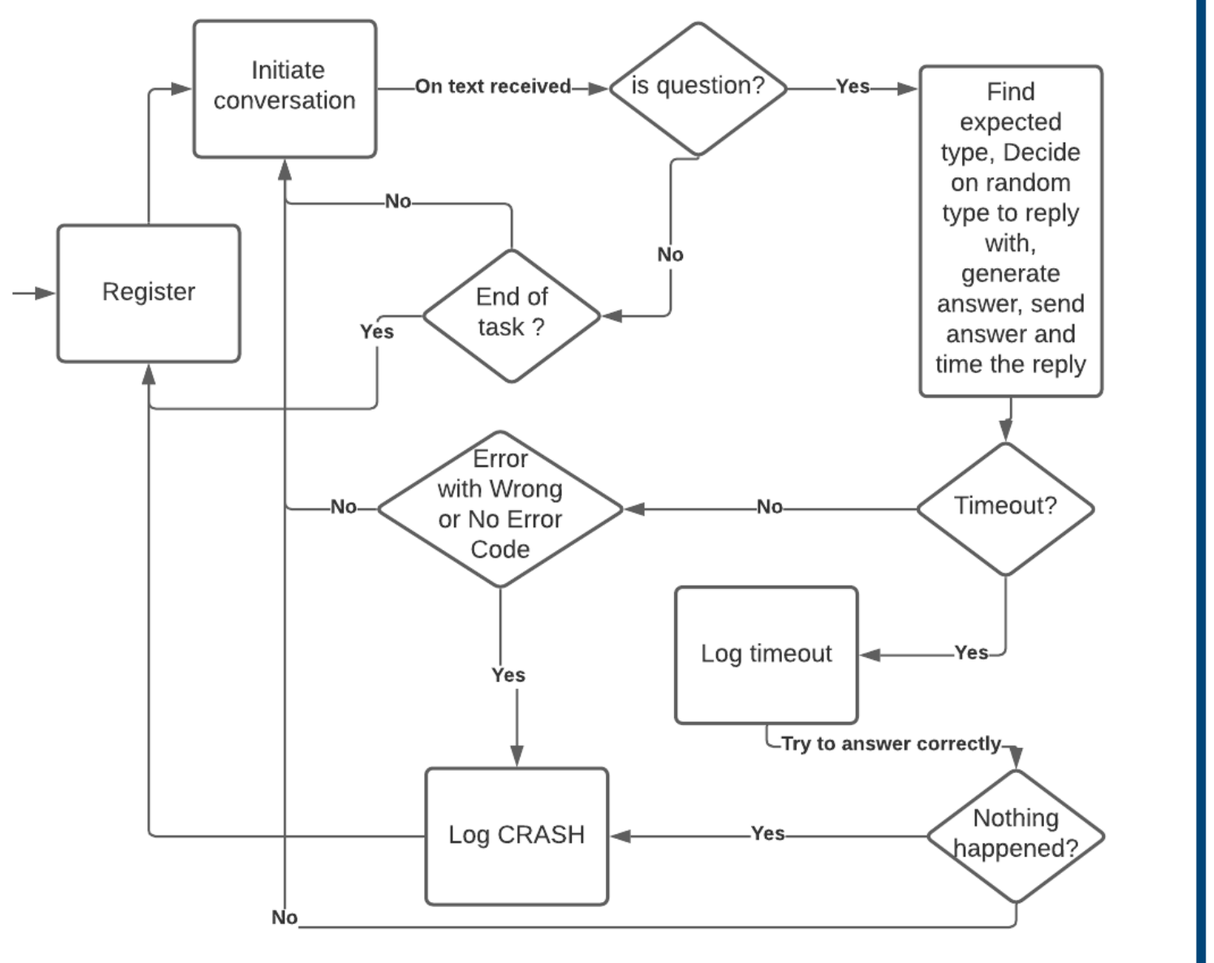

Figure 2: UML of the Robustness test

4. Dandelion Evaluation Methodology:

- Robustness: Input the wrong data type and ensure that the system handled the wrong input correctly and in addition classify results according to BALLISTA's C.R.A.S.H scale [1]. (Figure 2)
- Responsiveness: Simulate conversations of thousands of workers which generate random input as answers (Figure 1).

5. Ensuring Framework's Correctness:

Introduce bugs to Dandelion and verify that the testing framework is sensitive to such changes, the examples tested are:

- Force the test framework to only input the same random data type for all questions (Figure 3.3).
- Dandelion stops replying after a random number of questions (Figure 3.4).

Framework's Sensitivity to Changes in Dandelion

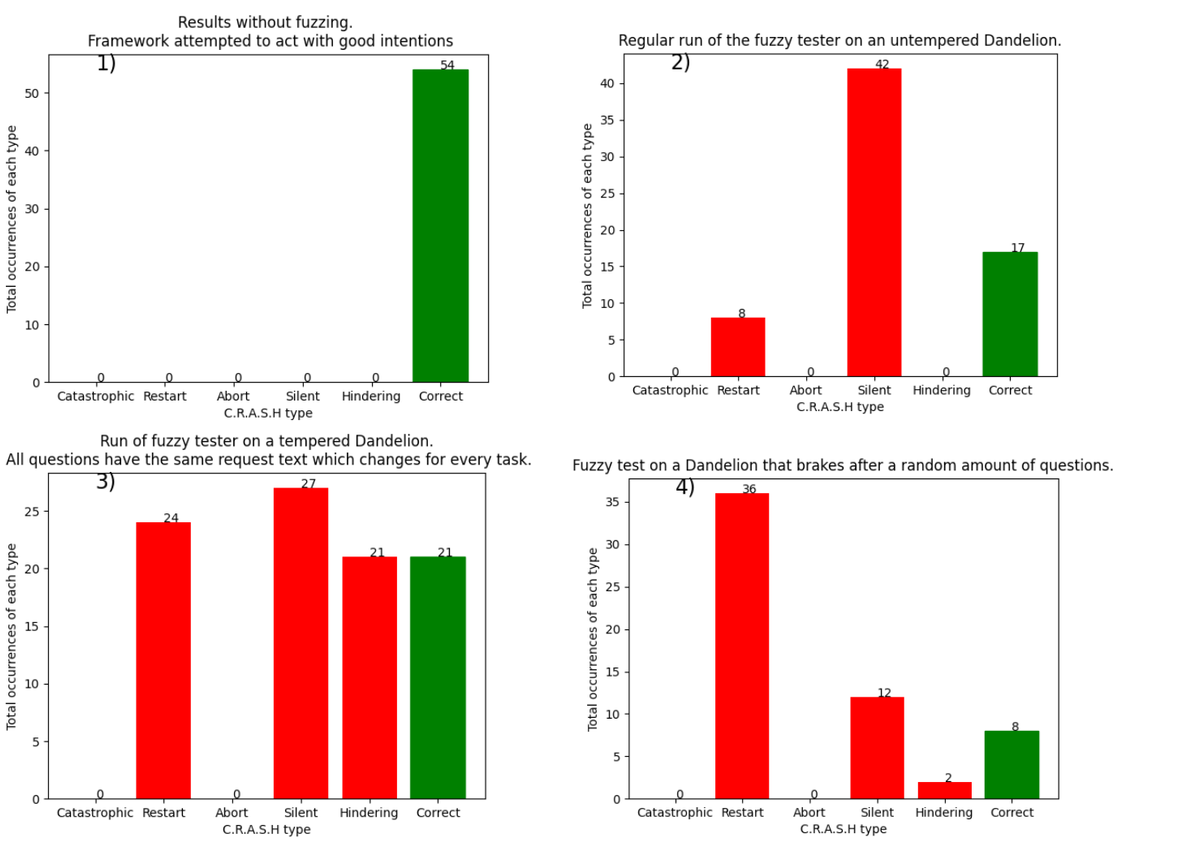

Figure 3: Results of robustness test on 4 different Dandelion versions

Simulation Results

Lower Bound estimation of the total time that is required to run a task

Mean (μ)	STD (σ)	#Users
5.788294135453459	0.9262387774394841	501

Table 1: Time needed to finish a task of 4 questions

Dandelion's Robustness

Result Type	Frequency
Correct	17
Restart	8
Silent	42
Hindering	0
Total	67

Table 2: Robustness results on regular Dandelion

6. Conclusions:

- Fuzz test is a good technique to test the robustness of a conversational crowd computing agent.
- The robustness test is sensitive to changes on Dandelion (Figure 3).
- Dandelion does not validate the user's input type (Table 2).
- The lower bound of running a task is 5.79 seconds (Table 1).

7. Future Work:

- Ensure that the users' input type is validated
- Ensure that the data committed to the database of Dandelion is validated.

References:
[1] Philip Koopman, Kobey Devale, and John Devale. Interface robustness testing: Experience and lessons learned from the ballista project. Dependability Benchmarking for Computer Systems, page 201–226.
[2] Iso/iec/ieee international standard - systems and software engineering – vocabulary.ISO/IEC/IEEE 24765:2010(E), 2010
[3] Facebook. Responsiveness requirements - messenger platform.

Contact Details:
Orestis Kanaris
o.kanaris@student.tudelft.nl