Lab 3 – Improvements Report

Yeganeh Vahabi
Course: DS

1. Fixes Implemented

- **Problem:** In the beginning, the main issue was that after saying a name such as "Tom", the system correctly recognized it and transitioned to `AskDay`. The inspector showed the expected state change, but the system never spoke the "On which day..." prompt. It appeared frozen, suggesting a SpeechState blocking problem.
- **Cause:** When the `RECOGNISED` event fired, the machine immediately transitioned to `AskDay` while SpeechState was still in an active `LISTEN` session. The microphone session had not fully closed, so when `AskDay` attempted to send a `SPEAK` command, SpeechState ignored it.
- **Solution:** The issue was resolved by avoiding transitions during an active listening session. Instead of moving to `AskDay` on `RECOGNISED`, the recognized value is stored and the machine waits for `LISTEN_COMPLETE`. Only after the listening session fully closes does the state transition occur, ensuring that `SPEAK` executes normally.

2. Identified Errors and Limitations

Limitation 1

- **Limitation:** Blindly re-asking the same question
  - When the user provides no response, I simply repeat the same question.
  - When their input fails grammar checks, I repeat the question again.
  - After many failed attempts (e.g., ten times), this leads to an infinite loop.
- **Solution:**
  - Introduce a retry counter in the dialogue context.
  - Enforce a maximum retry limit to prevent looping.
  - Replace exact repetition with a clarification prompt to guide the user.

Limitation 2

- **Limitation:** Duplicated confirmation logic
  - `ConfirmWithTime`
  - `Confirm`
  - `ListenConfirmWithTime`
  - `ListenConfirm`
- **Solution:**
  - Dynamically generate the confirmation utterance instead of hardcoding variants.
  - Use a single `Confirm` state.
  - Use a single `ListenConfirm` state.

Limitation 3

- **Limitation:** Repeating the same listening + retry + fallback pattern
  - `ListenPerson`
  - `ListenDay`
  - `ListenWholeDay`
  - `ListenTime`
  - `ListenConfirm`
- **Solution:**
  - Move shared event-handling logic to a parent state (refactoring).
  - Reuse the shared pattern for:
    * Person
    * Day
    * WholeDay
    * Time
    * Confirm

4. Remaining Limitations

- **No global handling for "no input"**
  There is currently no unified mechanism for detecting and responding to user silence across all states.
- **No out-of-scope handling**
  The system does not recognize or manage inputs that fall outside the domain of the dialogue task.
- **Hard reset when `confirm = false`**
  If the user rejects a confirmation, the system restarts the entire flow instead of recovering gracefully.
- **Grammar failure not distinguished from silence**
  The system treats unparseable input the same way as no input, preventing more nuanced error handling.