

Team 11

Architectural Thinking Assignment 7

1. Which entities and value objects are part of the business domain? List all relevant entities. Document at least 2 entities in detail and define the required value objects?

The relevant entities in our business domain are as follows:

- End User
- Query
- Result
- Response
- Speech Recognition
- Machine Learning
- Query Parse
- Context Info
- Feedback
- Third Party Skills
- Administrator

Now, let's understand the entities in detail:

- **End User:** These are the set of people who directly interact with our intelligent assistant. This entity forms an aggregate with Query, Response & Feedback entities. That is, users send different types of queries and expects appropriate response parameters to his/her input.
Value Objects: Username, Password, Location, Usertype, Language Preferences, Age, User ID
- **Query:** Queries are nothing but commands sent to the intelligent system by the corresponding users. Usually, these form an aggregate between End User, Feedback, Response, NLP Model, Query Parser
Value Objects: Query Type, Query Length, Query ID, Query Locale
- **Response:** Whenever the system is fed with queries, the input is parsed and an appropriate response is generated. This entity usually forms an aggregate with End User, Feedback, Query, NLP Model
Value Objects: Response Type, Response Locale, Response Locale, Speech Rate, Speech Accent

- **Feedback:** This entity comes into action during the context when, the user had received a response to his query from the system and then wishes to send a feedback as an input to the system to fine tune responses for the forthcoming iterations. This forms an aggregate between End User, Administrator, Query, Response, NLP Model, Query Parser & Context Info
Value Objects: Feedback Type, Rate, Locale, Feedback Length
- **Speech Recognition:** This entity handles the user inputs and outputs. The main task involved here is to handle natural language queries. So the usual aggregation is between End Users, Feedback, Query Parser, Query, Response
Value Objects: NLP Model, Locale, Rate
- **Machine Learning:** This is a crucial entity using which our system would be able to understand the inference between the predictors and responses. Further, make intelligent predictions using queries. We form aggregates between Query Parser, Query, Response, Context Info
Value Objects: Machine Learning Model Type, Model Data, Predictors, Response Type

2. Which domain events can change the state of the selected 2 entities?

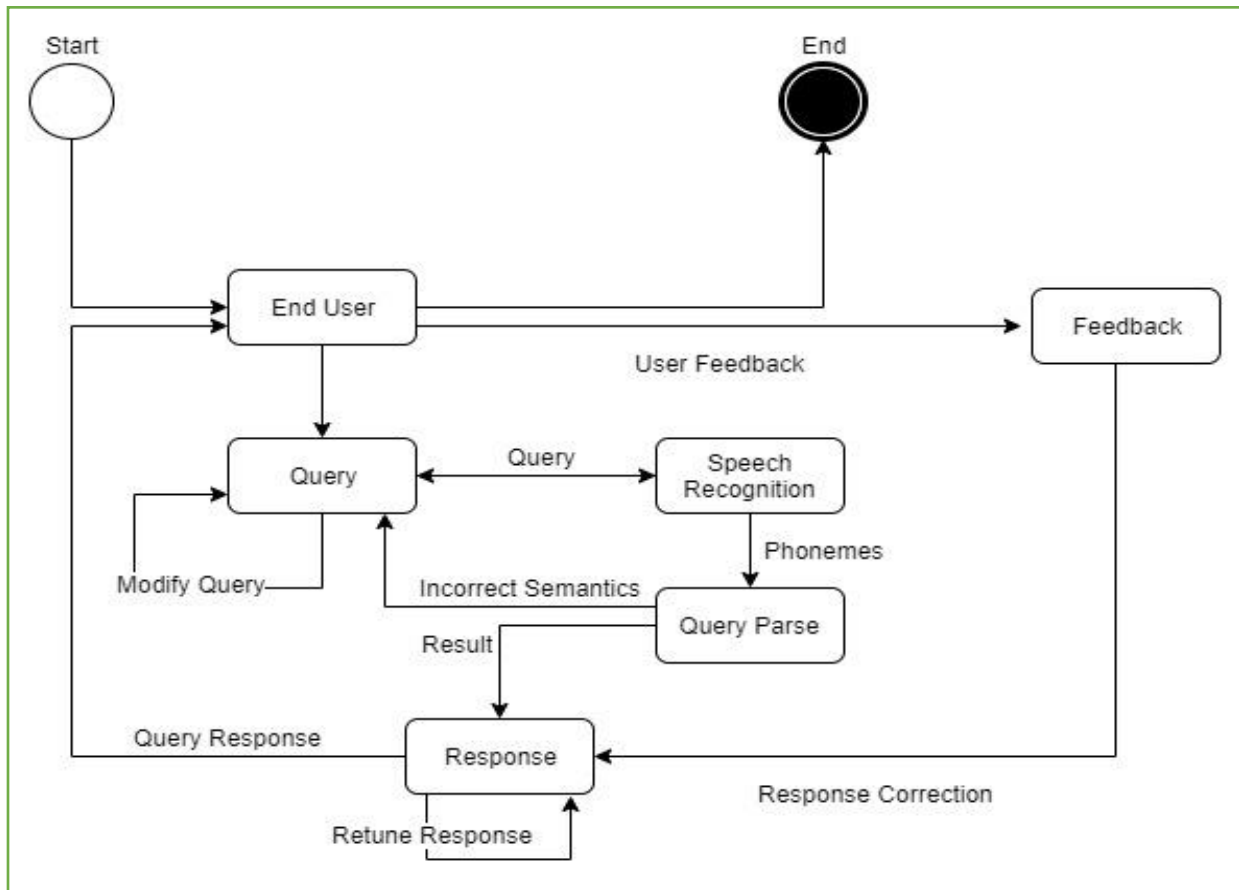
Let's choose the following entities to begin with: Query & Response

The domain events that change the state of the above 2 entities are: Query Sent, Query Received, Phonemes Recognize, Query Parsed, wrong Input Semantics, Input Locale Error, High Input Speech Rate, Breaking the input sentence, Response Locale Error, High Response Speech Rate, Query Incomplete, Incorrect Response, Input Noise

3. Model the lifecycle of the 2 entities showing all relevant states and state transitions. Think about things that can go wrong, potential rework loops, compensations, and error handling?

There is always the problem of 'Cold Start' in these intelligent assistant systems. But further, if there is no sufficient data to train the ML/NLP model. The burden increases on End User's/System owner's feedback. If the feedback submitted is incomplete or inaccurate, the system would perform worse over time and the accuracy of the response would suffer as the system continuously trains using incorrect data. The entities Speech Recognition & Query Parse takes the longest computational time in this scenario as there are lot of hidden aggregations and third party dependencies. Also, there might be potential loop between the Query & Speech Recognition entities. To reduce some delay or possible repeated control iterations, we can compensate the performance of the system by performing a tradeoff in input speech recognition & semantic accuracy of queries.

The feedback Machine Learning entity works as an error handling mechanism in our system i.e. the response from the system is sent back to the user. Based on the correctness, he can either finish the execution or submit correction required. Also, the errors can be reduced by performing hyperparameter tuning.



4. Which business services will your system offer? How do these services support the life cycle of the entities?

The system will provide the below services which will support the life cycle of the entities:-

- Accept Query:** The system accepts the query from the end user entity if and only if the input is semantically correct and unambiguous.
- Extract Phonemes:** Speech Recognition entity provides a service which is utilized to extract phonemes out of the audio query, if the input is in correct locale & within the acceptable speech rate.
- Check Query Semantics:** Query parser scans the input tokens/phonemes from Speech recognition entity for semantic correctness. If it is invalid, the user entity is notified to repeat the query again.
- Provide Contextual Response:** Once the query results are obtained from query parser entity, the appropriate response is sent back to the user via GUI/Voice. If the query parser

is unable to provide an accurate singleton result, the closest/related solutions are sent back as response.

- e) **Machine Learning Optimization:** The responses are fine-tuned for higher response accuracy by performing hyperparameter tuning. Further, it utilizes the contextual information from Context Store entity to supplement the result accuracy.
- f) **Provide third party skills:** If the user has signed-up, the system provides an opportunity for the users to plug-in external (ready to consume) skills. Which further improves the query parse entity capabilities & boost the service selection/variability of the system.
- g) **Context Parse:** As a cherry on the cake, this service improves the response entity's accuracy by 10 folds by providing additional context information to the machine learning entity.
- h) **Feedback Mechanism:** The system collects user's feedback to correct/improve the system's responses and further contribute in training the machine learning entity.

5. Do you want to add your entities to the same context or to different contexts? Sketch a simple version of a context map?

In our system, a more puzzling distinction might arise when the underlying concept is the same, but used in different ways, eventually leading to different models. That is, even though the query entity is same, it would probably be a conceptual mistake/unnecessary storing of all classes/parameters/value objects of an entity in a single domain. So we would want to add the entities to different contexts to prevent possible confusions!

Context Map:

