

## ***Step by step instruction:***

### ***How to create a LISSA dialogue?***

---

- In ***“start-lissa5.lisp”***
  - You need to load ttt source files from the path where they are located. For example our current code works on the HCI lab server and ttt files are located in the following path:  
"C:/inetpub/wwwroot/RocSpeakRafayet/ttt/ttt/src/load"  
You need to modify the path if it is needed.
  - You need to load all rules-files. Like the following:  
(load "rules-for-like-about-rochester-input.lisp")
  - There should exist a rules-file for each question LISSA asks (if you want LISSA to react to the user response). This rules-file contains choice packets for gist clause extraction from user replies along with a choice packet for creating the appropriate reaction to an extracted gist clause. We will consider it in more details in the following step.
- You need to actually create the ***“rules-files”***. Rules-files have four choice packets:
  - A packet for specific answers:
    - In the *\*specific-answer-from-...\** choice packet we basically look for some possible pieces of information around the question asked, for example for the major question we want to provide some gist clauses like “my major is ...”, “I am interested in ...”, “I have not decided on a major”, “I am a freshman” (they will add to the gist clause repository and later we need to use these gist clauses for generating an output)
    - After the gist clause, we should provide the subject of the gist. For instance “study-major” or “study-year”, etc.
    - You need to put “:gist” for the *\*specific-answer..\** choice packet. Also latency should be 0.
  - A packet for unbidden answers:
    - In the *\*unbidden-answer-from-...\** choice packet we look for related facts offered by the user that may preempt later lissa questions. For instance for “major” we could have “I find ... hard”. The subject of the gist should be provided as well. Like (study-major difficulty)
  - A packet for questions:
    - In the *\*question-from-...\** choice packet we look for potential reciprocal questions like “do you find ... hard?”, “what year of study are you in?”, etc. there is no subject parenthesis though.

- A packet for LISSA reactions:
  - \*reaction-to-...\* choice packet we consider the gist-clauses, so we basically match all of the above gist clauses and generate the proper LISSA output. Here the last parenthesis is (100 :out)
- In *"lissa5-schema.lisp"*:
  - You need to define \*lissa-schema" which is basically the main LISSA schema. This schema is a set of actions which by now could be of type say-to, reply-to, or react-to. Say-to actions make the LISSA outputs, reply-to actions get the user responses to a LISSA output and react-to actions make reaction as LISSA output. You need to carefully design the schema based on the conversation flow you anticipate to have.
  - You need to apply store-output-gist-clauses to all say-to actions. Here you assign a gist clause to every say-to output that LISSA generates, and later in "choose-gist-clause-trees-for-input.lisp" the program uses these gist clauses to select the proper gist tree by a pattern-matching.
  - You need to apply store-topic-keys to all say-to actions. Here you assign a topic key to what LISSA asks.
- In *"choose-gist-clause-tree-for-input.lisp"*:
  - you need to introduce the proper subtree for a recognized LISSA output gist clause. The LISSA output gist clauses which we declared in "lissa5-schema.lisp", will be matched here in order to select the right subtree; that is the subtree in which the input gist clause will be created.
- In *"choose-reaction-to-input.lisp"*
  - You need to modify READRULES for both \*reaction-to-assertion\* and \*reaction-to-question\*.
  - Under \*reaction-to-assertion\* all choice trees corresponding to LISSA questions should be covered. It is important that the matching keywords should be selected to match the corresponding gist clause, otherwise the reaction choice tree will not be evoked.
  - Under \*reaction-to-question\* all choice trees corresponding to the potential user questions (from LISSA) should be covered. The matching pattern is important and it should match the gist-question which we define in the rule files, otherwise the output will not be generated. (the questions that are covered here, could be both reciprocal questions that we declared in rule files, and other general question that is possible to be asked by a user)
  - Just note the READRULE for \*reaction-to-question\* generates output right here, but \*reaction-to-assertion\* calls a subtree for more pattern –matching.
- You do **not need to change** any of the following files:
  - "choose-doolittle-response-1.lisp"
  - "general-word-data-5.lisp"

- "lissa5.lisp"
- "choose-reactions-to-input.lisp"
- "schema-for-reactions-to-question+clause.lisp"
- "schema-for-reactions-to-answer-plus-question.lisp"