

Some questions have surfaced, and a student kindly submitted a list (below). My responses to the questions are in red.

This assignment requirements doc leaves a few holes in its explanation. The following ambiguities were noted and need to be addressed:

- What is to be the name of the third .pl file which will contain the 'gen_n1_reqs/0' and valid_req_spec/1 clauses?

Name the file 'meta.pl'.

- Are state, event and response names limited to any particular character set or maximum length restriction?

16 characters max, ASCII

- Are there any other expected natural language results/variations which should be output? The assignment requirements indicate 'the student will provide clauses in a separate file...to generate natural language from the requirements' indicating more than one clause is expected. However only one natural language output example for 'req/5' is given, indicating that only one main clause should be needed to generate the requirement output. Please reconcile this discrepancy.

Only one clause is necessary, but additional clauses may be written in support of the necessary clause.

- Can the requirement identifier be any value, or must it conform to the format id_00 (implying a fixed number of digits) or id_0 (implying a variable number of digits)?

Variable number of digits, max of 5.

- ☐ If the requirement identifier can be any value, is it limited to any particular character set or length restriction?

Max of 5 digits with three for the 'id_' sets the max id length to 8.

- ☐ If the requirement identifier must conform to the format id_00, should the system throw an error if it exceeded 100 (00 - 99) requirements?

Error checking will not be necessary.

- ☐ If the requirement identifier must conform to a fixed number of digits, do you wish to increase it to id_000 or id_0000 to accommodate a greater number

of requirements?

Not necessary to conform to a fixed number.

- Does the 'req/5' clause do the work of defining the requirement identifier from the value given? Or should the requirement identifier be validated as pre-defined somewhere else, or is it simply freeform?

The requirement identifier is intrinsic to the requirement fact, just as the identifier is intrinsic to the "state", "event", etc.

- Can/should requirements allow for negative terms? e.g. the example indicates "if the system is in a temp_normal state": should the 'req/5' clause be able to recognize "not temp_normal" and generate "if the system is not in a temp_normal state..."?

Only one clause is necessary, but additional clauses may be written in support of the necessary clause.

- The requirements specify that clauses should generate natural language, however the example for 'valid_req_spec/1' simply outputs a boolean. Should 'valid_req_spec/1' output natural language as well, e.g. "Req id_21: all terms for this requirement are valid."

valid_req_spec/1 should only evaluate to a logical true or false.

- 'valid_req_spec/1' validates 'req/5' which has 5 input terms. A single boolean output is not particularly meaningful; should 'valid_req_spec/1' report on the validity of each term individually? If so, what sort of (natural language or other) output should be generated to indicate this?

valid_req_spec/1 should only evaluate to a logical true or false. It will be used for support of other clauses.

- Is the 'valid_req_spec/1' clause intended to be run automatically, perhaps as part of gen_n1_reqs/0 output?

It may be used as part of your support for clauses. It will be used as part of the framework for assignment 2 and as support for my grading automation.