We have chosen not to use multiple-choice, but rather let the student formulate her own answer. To a certain question one may give many kinds of acceptable answers. In Sámi one may change word order, and also add many kinds of particles.

We use a ruleset file which disambiguates the student's input only to a certain extent, because there will probably be grammatical and orthographic errors. The last part of the file consists of rules for giving feedback to the student's grammatical errors, and rules for navigating to the correct next question of in the dialogue, due to the student's answer.

The system question and student answer are analysed together, delimited by the boundary marker ^qst. They first get a morphological analysis (left) , and are then disambiguated, and, if possible, assigned an error tag or a navigation tag.

Navigating by regular expressions

The system asks "How old are you?"- The answer is analysed, and a regular expression is used to read the answer. Dependent upon the age span, the user is then directed to dfferent follow-up questions.

```
  # Picking the age
MAP (&dia-adult) TARGET Num ("^1 QUE LINK 0 (Man_boaris_don_leat))
    (0 ("[2-9][0-9]"r))  ;
MAP (&dia-young) TARGET Num ("^1 QUE LINK 0 (Man_boaris_don_leat))
    (0 ("[1][0-9]"r)) ;
MAP (&dia-child) TARGET Num ("^1 QUE LINK 0 (Man_boaris_don_leat))
    (0 ("[1-9]"r)) ;
```

```
<utt type="question" name="How_old_are_you">
  <text>Man boaris don leat?</text>
  <alt target="young" link="Do_you_go_to_school_young"/>
  <alt target="child" link="Have_you_started_at_school_child"/>
  <alt target="adult" link="Do_you_work_adult"/>
  <alt target="default" link="Do_you_work_adult"/>
</utt>
```

Sentence generation

One of the main goals of the programs in OAHPA! is to practice language in natural settings with variation in the tasks. In order to provide variation in programs that involve sentential context we implemented a sentence generator. The sentence generator is used in the morphology in sentential context program (Morfa-C), and for generating questions to the QA drill (Vasta).

Suggestion: Either delete this or explain more.

```
<q level="2" id="go_ikte">
  <qtype>PRT</qtype>
  <question>
    <text>MAINV go SUBJ ikte</text>
    <element id="MAINV">
  <grammar tag="V+Ind+Prt+Person-Number"/>
  <sem class="ACTIVITY"/>
    </element>
    <element id="SUBJ">
  <sem class="HUMAN"/>
  <grammar pos="N"/>
    </element>
  </question>
</q>
```

Handling dialectical variation

When generating sentences or providing the correct answers for the user, we allow only normative forms in the chosen dialect. On the other hand, the live analyser used for the analysis of the user input accepts all correct dialect variants of the same grammatical word. We compile one normative but variation-tolerant transducer for analysing the input, and one strict one for each dialect for sentence generation.

In the source code, forms are marked as missing in certain dialects (the default being that all forms occur in all dialects). Below is an example of dialectal variation in the comparative inflection. The resulting transducers give *stuorát* for the KJ dialect and *stuorit* for the GG one, of the adjective *stuoris* "big".

```
+A+Comp:i%>X4b BUStem ; ! NOT-KJ
+A+Comp:á%>X4b BUStem ; ! NOT-GG
```