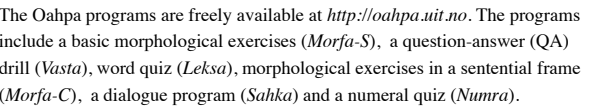


CAHPA!

<http://oahpa.uit.no>

We here present a set of interactive parser-based CALL programs for North Sámi. The programs are based on a finite state morphological analyser and a constraint grammar parser which is used for syntactic analysis and navigating in the dialogues. The analysers provide effective and reliable handling of a wide variety of user input. In addition, relaxation of the grammatical analysis of the user input enables locating grammatical errors and reacting to the errors with appropriate feedback messages.



The Oahpa programs are freely available at <http://oahpa.uit.no>. The programs include a basic morphological exercises (*Morfa-S*), a question-answer (QA) drill (*Vasta*), word quiz (*Leksa*), morphological exercises in a sentential frame (*Morfa-C*), a dialogue program (*Sahka*) and a numeral quiz (*Numra*).

The OAHPA! programs share a set of common resources: a pedagogical lexicon and a morphological generator that is used for generating the different word forms that appear in the programs. The dialectal variation is taken into account in the lexicon as well as in the morphology. The semantic class is used in the sentence generator for Vasta and Morfa-C. The lexical entry for *monni* "egg" is given to the right.

The OAHPA! programs share a set of common resources: a pedagogical lexicon and a morphological generator that is used for generating the different word forms that appear in the programs. The dialectal variation is taken into account in the lexicon as well as in the morphology. The semantic class is used in the sentence generator for Vasta and Morfa-C. The lexical entry for *monni* "egg" is given to the right.

If the user does not inflect the lemma correctly in the morphological exercises, she can ask for hints about the inflection, and try once more, instead of getting the correct answer straight away.

The detailed feedback messages are determined by the combination of morphological features in the lexicon and the inflection task at hand. The morphological specification below gives a rule stating that there is a vowel change in illative singular for bisyllabic nouns that end with the vowel *i*. The corresponding feedback message instructs the user to remember the vowel change.

The system-internal representation of *monni* states it is a bisyllabic i-stem, which triggers i > á change in illative.

The user types the erroneous *monnii*, and gets feedback from the machine.

A correct answer gets green colour as feedback.

The pedagogical programs in OAHPA¹ are based upon three pre-existing language technology resources developed at the University of Tromsø: a morphological analyser/generator, a CG parser for North Sámi and a number word generator compiled with the Xerox compiler xfst.

The main goal of the development of OAHPA! was to develop a language tutoring system going beyond simple multiple-choice questions or string matching algorithms, with free-form dialogues and sophisticated error analysis. Immediate error feedback and advice about morphology and grammar were seen as important requirements for the program.

The sentence generator in Morfa-C and Vasta is able to generate a virtually unlimited number of different tasks, and allows the student to use the programs over and over again.

Constraint grammar is a syntactic framework for choosing correct grammatical analysis of a given wordform, based upon the context it occurs within. Each rule removes or selects readings, or adds or removes a syntactic tag. Inappropriate analyses are removed, but the last analysis is never removed. CG thus always gives an analysis, and is therefore a very robust framework, well fit to handle potentially erroneous input.

The morphological analyser gives the words in *Makkár láibegálvuud don háliidat?* "What kind of bread do you want?" all possible grammatical analyses.

The CG grammar then picks the correct analysis, and adds grammatical function and dependency structure.

The overall evaluation shows that the students answer correctly slightly half of the time. By far the most popular program is the basic morphological drill (but the interactive programs have been logged for a couple of days only).

The 322 logged **Sahka** errors are distributed along the following lines:

For Sahka we test *precision* (correctly identified errors/all diagnosed errors) *recall* (correctly identified errors/all errors), and *accuracy* (correct judgements/cases).

Precision = 0.8; Recall = 0.68; Accuracy = 0.82 (N=584)

By using the syntactical analyser for North Sámi, combined with a set of error-detection rules, we have been able to build a flexible CALL resource. The programs are modular, and the modules may be improved by adding more materials words, tasks, dialogues, levels, words from textbooks. The CG parser framework was originally chosen as parser framework for Sámi due to its extraordinary results for free-text parsing. The present project has shown that CG is well fit for making pedagogical dialogue systems as well.

Antonsen, Lene, Saara Huhmarniemi and Trond Trosterud 2009: Interactive pedagogical programs based on constraint grammar. Proceedings of the 17th Nordic Conference of Computational Linguistics. *Nealt Proceedings Series* 4. <http://dspace.utlib.ee/dspace/handle/10062/9206>

The programs are based upon free-form interaction: Within certain limits, the student may formulate her own answer.

We use constraint grammar to disambiguate the student's input only to a certain extent, because there will probably be grammatical and orthographic errors. The manually written, context dependent rules are mainly used for selecting the correct analysis in case of homonymy. The last part of our grammar 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 merged and analysed together, delimited by the boundary marker *^qst QDL*. They first get a morphological analysis, and are then disambiguated, and, if possible, assigned an error tag or a navigation tag.

Above is a part of a dialogue in Sahka on

furnituring a flat. Below is the analysis of the third question and answer pair. The morphological analysis, is disambiguated and there are assigned a grammar-error-tag (<grm-missing-III>) and a navigation-tag (<dia-hivsset>):

The system may give feedback to grammatical errors. In the question which analysis is to the left, the systems asks “Where do we put the TV set?” The student answers *Moai bidje TV hivssegga* (“We put the TV in the WC”), with accusative *hivssegga* rather than the correct illative *hivssegii*.

[illegible]

The CG-rules disambiguate the input, and the rule above adds a grammar-error-tag (&grm-missing-III) to the input sentence (the analyse to the left). In the grammar feedback library, the tag in question looks up a message in the appropriate interface language (in this example, English), and the user is presented with the feedback *The answer should contain an illative*, as shown above.

```

graph LR
    MQ[machine question] --> Analysis
    UA((user's answer)) --> Analysis
    subgraph Analysis_Box [Analysis:]
        direction LR
        MA[morphological analysis  
(sme-norm.fst)] --> PP[post processing  
lookup2csg]
        PP --> DEI[disambiguating  
error detection,  
interpretation  
ped-sme.csg3]
    end
    Analysis_Box --> GF((grammar feedback))
    GF --> NI((navigation instruction))
    NI --> Analysis_Box

```

Navigating inside the dialogue is implemented in CG-rules. The user input is tagged during analysis with information on whether the answer is interpreted as affirmative or negative. In addition, a special tag indicates whether the sentence contains some information that should be stored for the following questions or utterances. The program is thus able to store simple information such as the student's name, place where she lives and for example the type of her car and use this information in tailored utterances.

In the example to the left the question is “In which room do we put the TV?” One of the alternatives for the navigation is due to the target tag being assigned because of the lemma *hivsset* (“WC”). The answer will be “That is not a good idea. Make a new try.” The CG-rule is:

MAP (Gale-Hispan) TACTIC CO. 3F 06 (prod.no.0101.77)
(1) (*known*) BAC110 BAC110

There are alternative links in the dialogue, one of them is due to the *&dia-hivsset* tag:

[illegible]

Every question has its own unique id, which is used in navigating between questions. In addition, the CG-rules may be tailored for specific questions, like in the rule above.

Age-tags are assigned with help of regex-rules to the answer to the question “How old are you?”. Due to the tags the system choose the correct link for moving to the next dialogue branch tailored to the student's age.