ViSiCAST Deliverable D2-3: Signing Tutor

Project Number:	IST-1999-10500
Project Title:	ViSiCAST
	Virtual Signing: Capture, Animation, Storage and Transmission
Deliverable Type:	Pub (Public)

Deliverable Number:	D2-3
Contractual Date of Delivery:	June 2002
Actual Date of Delivery:	December 2002
Title of Deliverable:	Signing Tutor
Work-Package contributing to the Deliverable:	
	WP 2 (Multimedia and WWW Applications)
Nature of the Deliverable:	PR (Prototype)
Author(s):	Hortensia Popescu, Sung-Eun Hong, Constanze Schmaling, Thomas Hanke, Rolf Schulmeister

Abstract:

This deliverable is one of several demonstrators showing how the ViSiCAST technology can be used in different application contexts. Whereas earlier demonstrators used motion capture technology for generating avatar movements, this demonstrator relies on synthetic signing as delivered in WP 5. This means that all signs shown in the demonstrator are described in the program with HamNoSys 4 and its non-manual counterparts, and are then transferred into SiGML to drive the animation.

This Signing Tutor prototype is designed for hearing learners of sign language for interactive use. This technology can visualise the productive sign language competence of the user and gives feedback regarding the correctness of a chosen sign phrase. Moreover, the versatility of the system offers the user the possibility to rotate the avatar and to zoom in and out in order to get a better view of the 3D movements in a sign.

The prototype consists of two units, number incorporation and directional verbs, two areas which cover sign language specific phenomena.

1 Introduction

This deliverable is one of several demonstrators showing how the ViSiCAST technology can be used in different application contexts. In contrast to earlier demonstrators, it does not employ motion capture technology for generating the avatar movements but relies on synthetic signing as delivered in workpackage 5. This means that all signs shown in the demonstrator are described in the program with HamNoSys 4 and its non-manual counterparts – as defined in D5-1 –, and are then transferred into SiGML (D5-2) to drive the animation (D4-2).

This Signing Tutor prototype is designed for hearing learners of sign language for interactive use. While typical multimedia programs for learning sign language provide scenarios, movies, or dialogues, more advanced learning programs offer interactive exercises to provide feedback to the understanding of the learner, the receptive language competence. None of these programs, however, can give feedback to the productive language competence of the learner. This signing tutor is the first technology that is able to visualize the sign language production of the learner and to give feedback regarding the correctness of a chosen sign sequence. Sign language learners can get feedback to their cognitive concept of signing in situations outside the classroom.

Using a tutor in learning has some advantages over a live seminar:

- The student can study according to her individual pace.
- The student can study independent of time and location.
- The student may repeat tasks as often as needed or wanted.
- The system is able to assist the memory of the learner whereas the information in a seminar is difficult to memorize.

The same advantages are offered by using a CD-ROM. However, using an avatar for learning instead of stored video sequences on a CD-ROM has additional advantages:

- The storage space on a CD-ROM is limited. The avatar can generate an unlimited amount of signs and signed phrases.
- The student can produce any sign or signed sequence with the help of the avatar, i.e. there is no need to foresee what the student creates and to store all possible phrases as videos on the CD-ROM.
- The student may use the avatar to test her own productive language competence.
- The student may even test herself by letting the avatar make mistakes.

Moreover, the advantage of using an avatar is its versatility: The student is able to rotate the avatar and zoom in and out as necessary in order to better recognise the 3D physical movements constituting the signs.

This tutorial consists of two units, number incorporation and directional verbs. We have chosen these two areas because they cover sign language specific phenomena and because there already exist teaching materials for both that we could use as a basis for our tutorial.

2 Structure of the tutorial

The tutorial starts with an introduction by the avatar (Visia): “Hello, my name is Visia, I am an avatar. We will now do exercises together, for DGS number incorporation and bidirectional verbs. When you go to the next page you will see two boxes, choose one to continue.” A written translation of the signed sequence is simultaneously shown on the screen so that the student can get used to Visia.

On the next page, the student can choose which unit she wants to go to first.

Each unit starts with a text in written German, which introduces the student to the grammatical features of the unit.

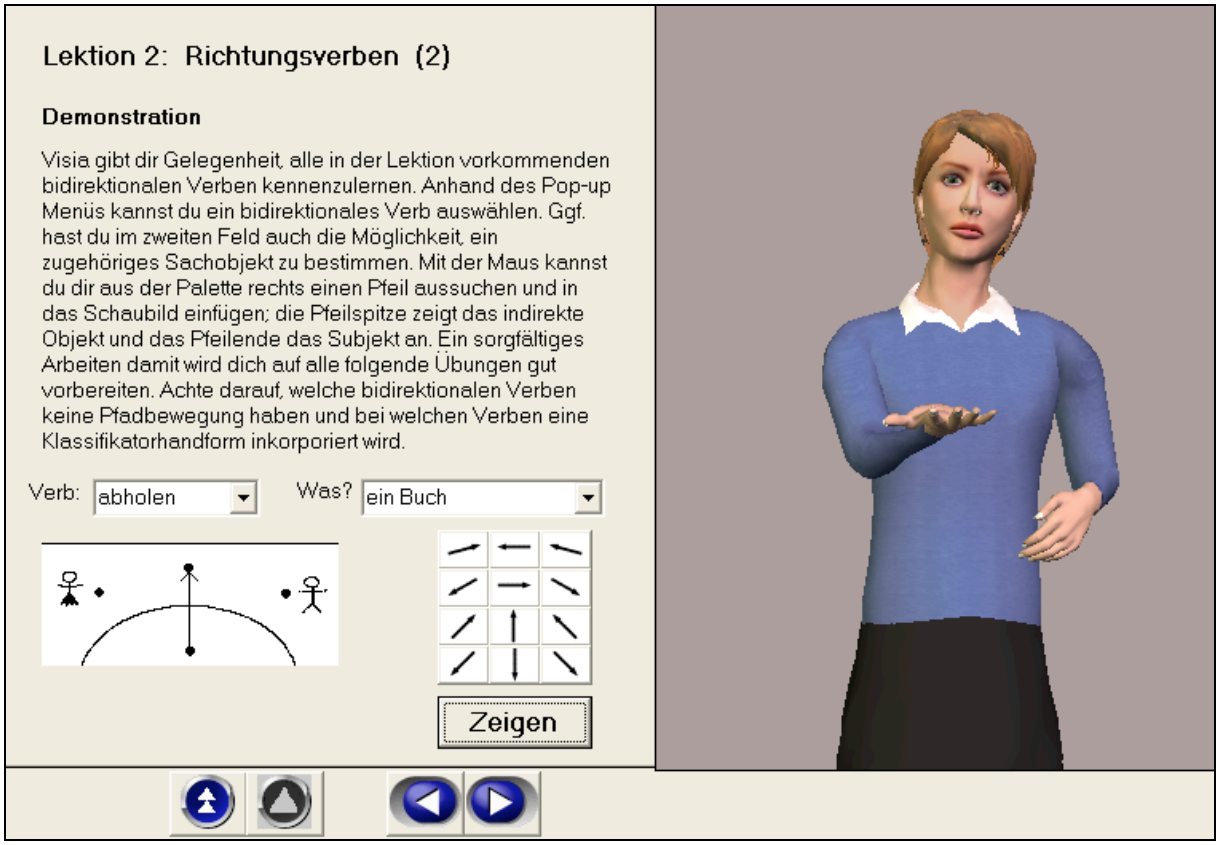
The following page, called “demo” page, gives the student the opportunity to practice new or already existing knowledge. The student can input as many example phrases as necessary and ask Visia to perform them.

2.1 Unit 1: Number incorporation

In Unit 1, one-to-two-word sequences are used. Some signs in DGS allow number incorporation for the numbers 1–10 (e.g. HOUR, WEEK)—these are one-word phrases; in other signs number incorporation is not possible (SECOND, DAY). The lexicon consists of the ordinal numbers 1–9999, cardinal numbers, and the signs SECOND, MINUTE, HOUR, O’CLOCK, DAY, WEEK, MONTH, YEAR, EURO, and CENT.

2.2 Unit 2: Directional verbs

Unit 2 covers the group of fully directional verbs, including the so-called orientable verbs. There are different types of directional verbs: Some take two arguments, others take three arguments; some can incorporate a classifier handshape while others cannot. The spoken language sentences therefore consist of three or four slots. The lexicon comprises of 30 directional verbs and six objects (BOOK, TABLE, LARGE-BOWL, PAN, BOTTLE, A-PIECE-OF-THREAD).



Once the student has finished practicing with the avatar, she can go to the following page where she can choose between the different exercises.

3 Exercises

In each unit, there are four different types of exercises with different types of interaction. Exercise 5,“tutti frutti”, generates exercises of the four different types in random order. All exercises are interactive: The student has to either select the correct answer, type in a correct answer herself, or judge whether a given answer is correct or not.

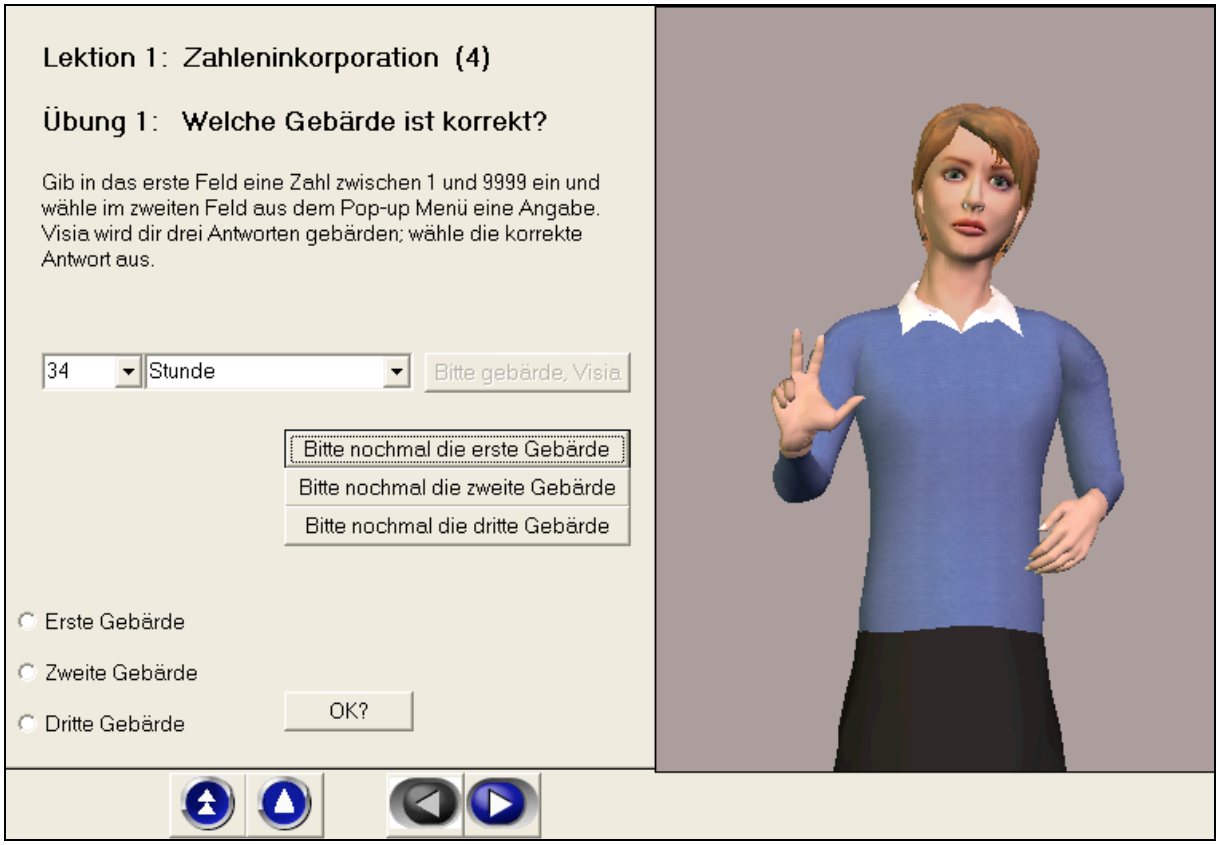
In the following sections, the different types of exercises will be described. The description will be illustrated for Unit 1; exercises in Unit 2 are structured in the same way.

3.1 Exercise 1: "Which signed sequence is correct?"

In Exercise 1 in Unit 1, the student can enter two words: In the first field, a number has to be typed in; the second field is a pop-up menu with the above-mentioned lexical items. Visia will sign three different answers, one correct answer and two wrong answers. The student has to select the correct answer. The answers look fairly similar, e.g. they contain the same number but a different object; or they contain a number that is performed in a similar way. Visia also gives so-called "pseudo-correct" answers, e.g.:

- In two-handed number signs, the hands move in opposite direction instead of moving in the same direction.
- In some signs, Visia uses the wrong palm orientation, e.g. in the sign WEEK, the palm faces upward instead of facing downward.
- Visia incorporates numbers in signs that do not allow incorporation.

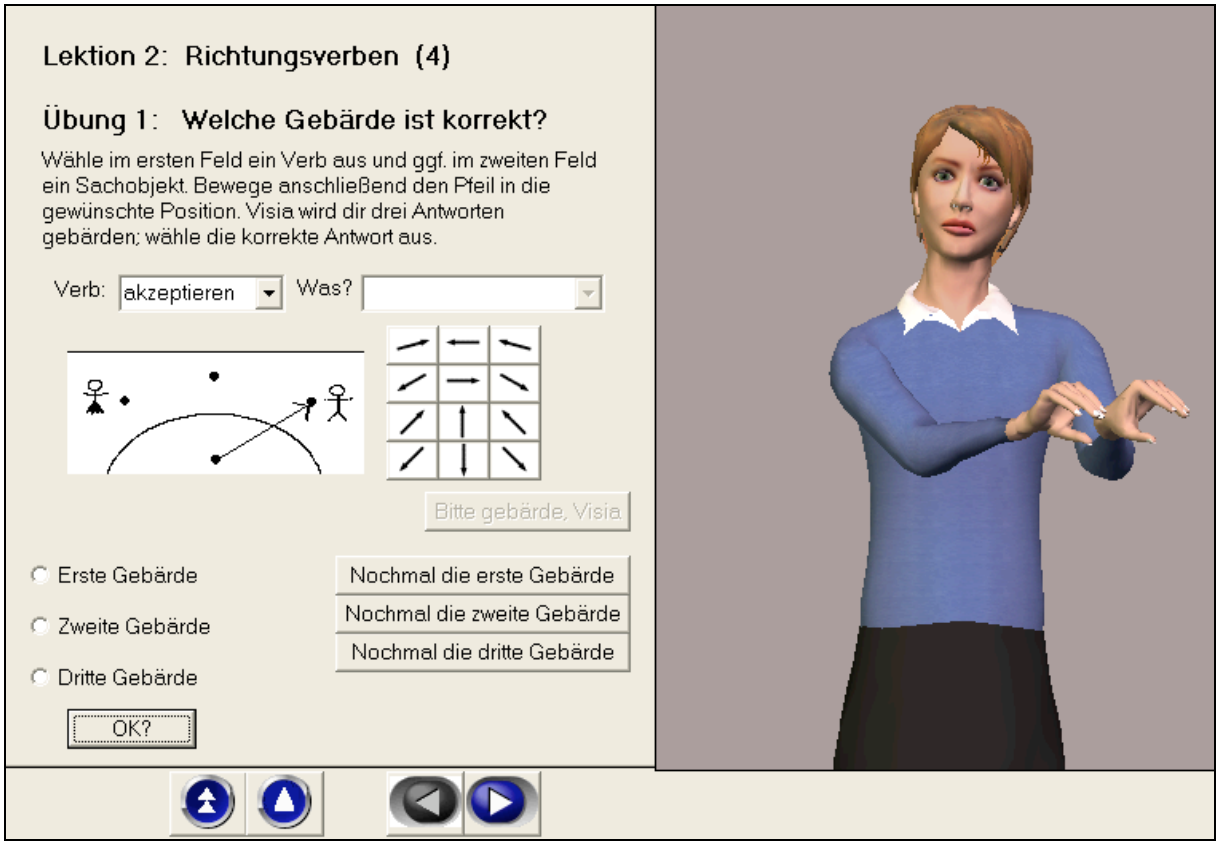
The student can ask Visia to repeat each of the signed sequences.



In Unit 2, the student can compose a sentence that consists of maximally subject, direct object, indirect object, and verb. She can select a verb from a pop-up menu and can determine subject and (in)direct object by choosing the corresponding arrow from an arrow table. This arrow will be inserted into a drawing with four locations in signing space for 1st, 2nd, and two 3rd persons, located on the left and right side in space. If the student has selected a verb that can take both direct and indirect object, another pop-up field opens in which she can select the direct object. Visia then signs three different answers.

Visia's pseudo-correct answers in this unit are generated using modifications of the different components of the verb descriptions, for example:

- The direction of movement is wrong, e.g. reverted.
- The classifier handshape is incorrect.
- Visia incorporates a classifier handshape in a verb sign that does not take a classifier.



3.2 Exercise 2: "What does Visia sign?"

In this Exercise, Visia signs one or two signs that are randomly generated. The student has to compose her answer by typing in a number from 1–9999 and choosing an object (SECOND, MONTH, etc.) from the pop-up menu in Unit 1, or choosing a verb, the correct arrow and possibly a direct object in Unit 2. As in Exercise 1, Visia can be asked to repeat the signed sequence.

3.3 Exercise 3: "Which text answer is correct?"

Visia signs a phrase that is randomly generated. The student is given three possible answers and has to select the correct one. The answers are fairly similar in order to make the choice more difficult. If necessary, Visia can repeat the sign(s).

3.4 Exercise 4: “Correct or wrong?”

While Visia signs one or two signs a translation is simultaneously inserted. Signing and text may or may not match. The student has to decide whether Visia gives a correct translation, i.e. whether the signing is correct and corresponds to the German text. The student can choose between three different answers:

- Correct
- Wrong because text and signing don’t match.
- Wrong because the sign is not performed correctly. (In this case, Visia signs a pseudo-correct answer, see Exercise 2 above).

The student can ask Visia to repeat her performance.

3.5 Exercise 5: “Tutti frutti”

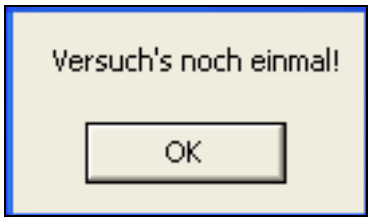
A random selection of exercises 1–4.

4 Feedback

After the student has selected her answer, Visia gives feedback by responding with either “BRAVO”, “GREAT”, “CONGRATULATIONS”, or with “NO”, “WRONG”, “WHAT A PITY” which appear in random order.

At the same time, for each correctly answered question the student gets a green marker at the bottom of the page. For incorrectly answered questions a red marker appears. Maximally ten markers are visible, they shift to the left.

If an incorrect answer is given, a message box appears on the screen asking the student to try another answer.



In Exercise 2, if the student's second try to answer is incorrect as well, Visia will show the correct answer. Visia signs "CORRECT ANSWER", and the correct answer appears in written text in a message box.

The tutor produces new exercise tasks ad infinitum so it is left to the user to decide when to leave an exercise to continue with another unit. In order to encourage the student to explore the program, the marker feedback mechanism is not even explained by the system. It is expected that users understand the concept from what they see, but feel free to stay in a particular exercise as long as they consider necessary.

5 Navigation

On each page there are four buttons for navigation:

"Home": Go back to the beginning of the tutorial.

"Up": Go back to the exercise selection page in this unit.

"Backward": Go to the previous page.

"Forward": Go to the next page.

Buttons with inappropriate targets in a certain context are inactive and greyed out. Whereas the Forward button in the standard element enables the user to continue work with the tutor, the Up and Home buttons are only used to terminate a certain task.

6 Implementation issues

The implementation of the signing tutor consists of three major components:

- user interface implementing the learning environment
- linguistic processing
- control of the avatar.

In order to be able to show with this proof of concept that a sign learning environment can be implemented in a compact piece of software usable off-line without hardware or system software requirements exceeding those required for running the avatar, the tutor was implemented from ground up in Microsoft VisualBasic.

The use of the implementation language makes it feasible, however, to integrate the components into other environments such as eLearning shells or Web browsers.

6.1 User interface

The main aim of the user interface design was to reduce the time needed to learn the navigation to a minimum. Therefore, only standard user interface elements, such as buttons and menu boxes, were used. This also holds true for the selection of actor-addressee relations for directional verbs: here, directly clicking onto the signing space map might be a more intuitive user interface for people using the tutor over a longer period of time.

6.2 Linguistic processing

For the linguistic processing, the decision taken meant that the existing HPSG environment could not be integrated into the tutor as this system itself has substantial system requirements. Instead, those components of the HPSG system needed for implementing the two units of the tutor were reimplemented in VisualBasic.

For the introductory signing as well as for number incorporation, the sign manipulation is straightforward: Signs are either completely specified as HamNoSys token strings[1] plus mouth pictures or as such strings containing parameters. Parameter substitution takes place governed by some simple rules, as can be seen from the following pseudo code:

```
Function ComputeNumberIncorporationHNST(theNumber,theObject)
hnst = ""
If 0<theNumber< 11 And CanIncorporate(theObject) Then
    NumberIncorporation (theNumber)
```

[1] The HamNoSys token format is a bracketed structure created for the compact representation of sign chains in the natural language generation context. Each element of the chain consists of an optional gloss, the HamNoSys description of the manual part of the sign as well as the tagged representation of all nonmanual parts of the sign. The HamNoSys description is given by means of tokens, i.e. names for each HamNoSys symbols. The tutor uses the nonmanual part only to describe mouthing, the encoding is described in D5-1.

```
    SignObject (theObject,hnst)
Else
    SignNumber (theNumber,hnst)
    NumberIncorporation (1)
    SignObject (theObject,hnst)
End If
return hnst
End Function
```

`CanIncorporate` makes a dictionary-lookup to decide whether the sign allows for number incorporation or not. `NumberIncorporation` sets the parameters (handshape and handedness) that feed into `SignObject` which simply looks up the dictionary entry for the object to be signed and substitutes the parameters if any. It then appends the sign for the object to hnst which might already contain the number sign looked up and inserted by `SignNumber`.

For the unit on directional verbs, a more complex approach is needed. Therefore, the module computing the form of directional verbs mimics the HPSG approach as far as possible (cf. deliverables D5-1 and D5-3). Lexical entries are automatically converted from HPSG TDL format to Basic statements by means of a Perl script:

```
Case "schenken" 'GIVE-A-PRESENT
    mouth = "SENk"
    ndh = "hamsymmpar"
    hsh = "hamfinger2,hamthumboutmod,hamfingerbendmod"
    plm = "hamparbegin,hampalml,hamplus,hampalmr,hamparend"
    efd = "hamparbegin,hamextfingerol,hamplus,
             hamextfingeror,hamparend"
    srcheight = "hamchest"
    srcdist = "close"
    golheight = "hamchest"
    goldist = "close"
    pthmod = "hamarcu"
    fob = False
```

In the case of classifier verbs, these structures once again contain parameter values to be instantiated from classifier descriptions as selected by the objects to be involved:

```
Case "geben" 'GIVE
    mouth = "ge:b"
    ndh = "#1"
    hsh = "#1"
    defaulthsh = "hamflathand,hamthumboutmod"
    efd = "#1"
    defaultefd = "hamextfingerl,hamorirelative"
    plm = "#1"
    defaultplm = "hampalmu"
    srcheight = "hamchest"
    srcdist = "close"
    golheight = "hamchest"
    goldist = "close"
    pthmod = "hamarcu,hamsmallmod"
    fob = False

Case "cl-manip-thin-rectangle-vertical"
    clhsh = "hamceeall,hamthumbopenmod"
    clefd = "hamextfingeror"

Case "ein Buch" 'BOOK
    Classifier "cl-manip-thin-rectangle-vertical"
    return "[[BUCH],[mbu:x],[hamsymmlr,hamflathand,
            hamextfingero,hampalml,hamtouch,hamparbegin,
```

```
                hammover,hamsmallmod,hamreplace,
                hampalmu,hamparend]]"
```

The construction of the whole phrase is then accomplished by the following code:

```
If Has3Args(Verb) Then
  hnst = AddTheme(Theme) + "," 'also instantiates the parameters from
            classifier defined by theme
  Else
    hnst = ""
    Classifier ""
  End If
  If verbDictionary(Verb) Then
    hamnosys = BuildHamNoSys(verb,subject,addressee)
    hnst = hnst + "[[" + UpperCase(Verb) + "],
                         [m" + mouth + "],
                         [" + hamnosys + "]]"
    end if
    return "[" + hnst + "]"
  End If
```

Of this, `BuildHamNoSys` is by far the most complex component, responsible for deriving the directional form of the sign from the lexicon entry as shown above.

6.2.1 Generation of wrong answers

In order for Visia to be able to give wrong answers (in the exercises 1 and 4), we implemented a function which replaces certain signs in the correct sequence with other similar signs, eg: YEAR is replaced with O'CLOCK, the numbers up to 10 are replaced with (number+10), and in the Unit 2, the SRC and GOL (source and goal) of the verb are reversed or the verb is replaced with a similar one.

6.2.2 Lexical entries created for the pseudo-correct answers

New lexical entries were created in order to generate the pseudo-correct answers. This was done by modifying the value of one feature in the description of the sign. For example, the pseudo-correct entry for the verb SCHENKEN ("to give a present") was created by modifying the values of the features PLM (palm orientation) and EFD (extended finger direction).

```
Case "schenken_pseudo"
    mouth = "SENk"
    ndh = "hamsymmlr"
    hsh = "hamfinger2,hamthumboutmod,hamfingerbendmod"
    plm = "hampalml"
    efd = "hamextfingerol"
    srcheight = "hamchest"
    srcdist = "close"
    golheight = "hamhead"
    goldist = "close"
    pthmod = "hamarcu"
```

6.3 Avatar control

In order to completely encapsulate avatar control, a new ActiveX control was developed by UEA (Ralph Elliott). This basically provides one single entry point, `playHNST`, taking the HamNoSys token format as input, internally translating it into SiGML (D5-2) and then using that to drive the avatar. It provides the user interface functionality of the avatar display, i.e. rotation and zooming of the avatar in response to mouse events. In addition, it resizes the avatar in response to the user resizing the container window.

This control cannot only be used from within VisualBasic, as in the current implementation of the tutor, but also, for example, in web-based environments.

7 Implementation deficits

For stability reasons, the tutor still uses the last pre-TNG version of the avatar. As a consequence, non-manual behaviour implemented by the tutor does not show. This does not only concern mouthing, but also Visia's feedback to user input, where is might be especially disturbing that Visia does not show any facial expression.

The number incorporation generation occasionally shows weaknesses in the avatar control with respect to collision detection between the hands and the face. However, most of these problems could be avoided by manipulating the lexicon entries for the signs involved.

In the case of directional verbs, inverse kinematics that is part of the avatar control shows some weaknesses only occasionally found before. This is not too surprising as directional verbs are one area in sign language that fully exploits the signer's wrist motion. Only when more constraints on wrist movements, including those depending on the handshape, have been implemented, it will be feasible to set up test suites in order to balance motion between the arm limbs and the shoulder.