Version

1.2

THE OMEGA-IS PROJECT GROUP

Mikael Heimann, Mats Lundälv, Tomas Tjus & Keith E. Nelson

Omega - interactive sentences Manual

OMEGA- is

Manual version 1.2

www.omega-is.com

Developed In association with Center for Child & Adolescent Mental Health, University of Bergen, Norway, Department of Psychology, Göteborg University, Sweden and DART, The Queen Silvia Children's Hospital, Göteborg, Sweden, Department of Psychology, Penn State University, USA.
Translation: Åsa Mühlenbock and Mats Lundälv

Content

CHAPTER A: INTRODUCTION 1

WHAT IS OMEGA–IS? 1
GETTING STARTED, AND WHAT IS AN APPROPRIATE LEVEL? 2
CREATE 3
B-lessons *3*
D-lessons *3*
F-lessons *3*
H-lessons *3*
J-lessons *3*
L-lessons *3*
INSTALLATION 4
START THE PROGRAM 5
MOTIVATION AND SUPPORT 6
ADAPTABILITY 6

CHAPTER B: CREATE 7

CREATING SENTENCES AND EVENTS 7
REGISTER A NEW USER (PUPIL) 7
CHOOSING A LESSON 9
LANGUAGE VERSIONS 10
LINGUISTIC MATERIAL 10

CHAPTER C: TESTING 13

WHY TESTS? 13
TESTING IN THE LEARNER (PUPIL) MODE 13
TESTING IN THE TEACHER MODE 13
HOW TO HANDLE LEARNER/PUPIL SETTINGS 14
VIEWING RESULTS 16
ANALYSING RESULTS 17

CHAPTER D: EDITING LESSON MATERIAL – THE LESSON AND ANIMATION EDITORS 18

OVERVIEW 18
EXERCISES – CONTENT AND STRUCTURE 19
TWO LESSON EXAMPLES 20
Example 1: A lesson with video sequences 20
Example 2: A lesson with Omega animations 22
THE FILES AND FOLDERS OF OMEGA-IS 26
The installation folder of Omega-is – an overview 27
*The "**anim**" folder: 27*
The "lesson-en" folder - English lessons: 28
The "media" folder: 28
The "register" folder – pupil settings and data 29
SUMMARY - CONCLUSIONS 29
1. A lesson based on video sequences 29
2. Lessons based on Omega animations 29

CHAPTER E: THE LESSON EDITOR 31

OVERVIEW 31
TWEAKING EXISTING LESSONS 31
Example 1: Remove a word/item from an existing lesson: 31
Example 2: Add a word/item to an existing lesson 33
CREATE NEW LESSONS 34
A new three item lesson – existing material 34
A new three item lesson – new material 36
RECORDING NEW WORDS/PHRASES 37
MAKING NEW LESSONS AVAILABLE 38
ADVANCED – VARIABLES, STORIES ETC. 38
Variables - references 38
Linked animations 39
Stories 39
FUNCTIONS AND COMMANDS 41

CHAPTER F: THE ANIMATION EDITOR 45

OVERVIEW 45
MINOR CHANGES IN EXISTING ANIMATIONS 46
Adding new actors to an animation 46
Changing the length of animations 48
Changing the background image 49
Changing the positioning of the animation path – in the scene 50
Changing the shape of animation paths – in the scene 51
CREATING SIMPLE NEW ANIMATIONS - BASICS 52
Starting the Animation Editor 52
Selecting the background 52
Adding and adjusting 'wings'/'props' 53
Adding and adjusting actor paths – on stage 55
Adding and adjusting new actors 56
Adjusting paths and events on the TimeLine 57
ADVANCED FUNCTIONS 59
Internal animation 60
Using variables 60
Tips and tricks 62
FUNCTIONS AND COMMANDS 63
IMAGE FORMATS 66
Backgrounds: 66
Actors 66
Wings/props 66

CHAPTER G: EDUCATIONAL PERSPECTIVES 67

MIR – ABOUT OUR METHOD 67
The computer as a linguistic prop 67
DEVELOPMENT OF LANGUAGE SKILLS AND TEACHER–LEARNER INTERACTION 68
Fundamental factors 68
Necessary factors 69
Facilitating factors 70
L.E.A.R.N: SOME MORE THEORY 71
ON THE IMPORTANCE OF CONVERSATIONS: RECASTING 73
Recasting - how to do it 74
RESULTS FROM RESEARCH 75
SELECTED BIBLIOGRAPHY 76

Chapter A

Chapter A: Introduction

What is Omega – is?

Omega-is has been created to stimulate the development of language and communication in learners beginning to explore the mysteries of the written language. What do the letters represent? How can they represent words and events? The language matter of the program is meant to be explored by the learner with help from - and in interaction with – a teacher or parent. The language material and the appended animations do not only offer motivational literacy training but also give occasion for conversations where the learner can express his or her imagination and thoughts

The program consists of two main sections. The *lesson section* (figures 1 and 2)comes with language and text material where the learner can create events with the help of text buttons. The learner can also test his or her proficiency by first viewing the event, then choosing words and creating the sentence that best represents what he or she has just viewed.

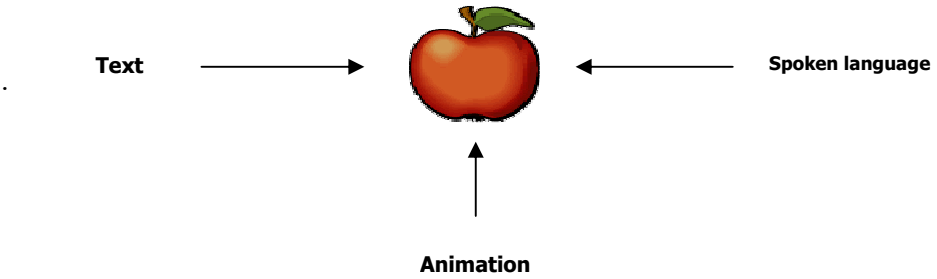


Figure 1: *Lesson section content*

In the *editing section*, the adult can edit the existing material or create entirely new materials in order to offer the learner an ultimately well-adapted and stimulating learning situation

The multimedia lesson material has been created according to the following structure (figure 2). The language content is represented by text. Having clicked on the word, the learner will hear the word by means of a recorded voice and then view the word as a picture or, if it is a phrase, as an event. Thereby, the learner – in a limited-time window – is given support in three different forms (text, speech,

animation) which will facilitate his or her understanding of the concept of lingual representation.

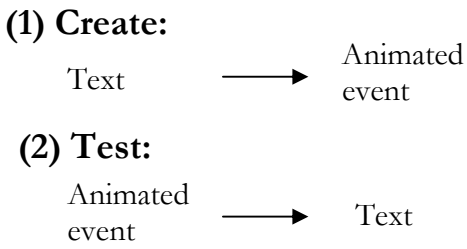


Figure 2: *The program design*

After starting up the program, the name of the learner is typed into the name list. Then the learner can start exploring the language material in *Create* or test his or her reading ability in *Test*.

Omega-is

- **includes an extensive language material:**
  - **29 basic lessons with around 200 words (plus 3 stories and 16 demo lessons)**
  - **provides sufficient material to create 1936 sentences (plus the content of stories and demo lessons)**
  - **average level is 3-word s-v-o sentences**
  - **most advanced level is short stories**
- **has an open design**
  - **i.e. the option to add new language and media material over time**
  - **such additional content will be provided by the developers (on the Omega-is web site www.omega-is.com), but may also be added by teachers or parents**

Getting started, and what is an appropriate level?

How much did I learn, and how much did I know before? These matters can be explored in the **Pretest** section (See chapter C "Testing in teacher mode"). This means that the learner can find out how much he or she knows before starting to work on a lesson. It also gives the adult working together with the learner an opportunity to establish the appropriate level on which to start working. A rule-of-thumb here is to use 20% correct as a starting point, as the learner should have some previous knowledge as well as the challenge to learn something new. However, this per cent figure should be viewed as a rough measure, since our experience shows that some learners, despite scoring lower than 20% correct on the Pretest, still learn the lesson content. It is important to find balance between the pleasure of the learner in exploring the material and the level of difficulty (a level of too great difficulty will only cause frustration).

Create

The central part of the program is the Create activity. The learner explores the program, getting acquainted with text and how to use new language skills. The contents of the language material:

B - lessons

This section contains one-word material only, the idea being that the learner should understand how the written word represents a concrete object, i. e. only nouns are represented. The simple structure can also make learners who, due to motor disabilities, have difficulties to work on the computer gain insight into how to illustrate a text by clicking. The words have been selected in part because they compel a phonological (audible) and/or orthographical (pictorial) difference, such as two words of great similarity : "Hound, House".

D - lessons

These consist of two-word sentences with a noun and a verb to illustrate the notion that somebody does something and how simple grammar is built by a subject and a predicate.

F - lessons

Three-word sentences with subject, verb and object (S-V-O). Some words from previous lessons reappear, but there are also new words for additional phonological drilling. Words are sometimes placed according to a strict grammatical order where subject-predicate-object have given columns but sometimes also with nouns only in one column and verbs in another. The selected sentence will always, however, have a grammatically correct structure.

H - lessons

The most common prepositions are represented in order for the learner to explore situational positioning. Use of the genitive is also practised in certain lessons as learners explore phrases with a possessive content (such as "the panda's table").

J - lessons

Contain long sentences with added new words. The purpose is reading practice combined with exploring the limits of the learner's working memory which is important when it comes to reading and comprehending of longer texts. This may also be checked in the testing part if the learner likes this type of challenge.

L - lessons

Using different alternatives, the learner can create his or her own story with different types of principal agents and scenarios, such as "Where is the action going to take place?", "What is the main character looking for?", "Who can help?" The purpose here is to give the learner an understanding of how stories are created, to encounter additional complex sentences and to provide the learner with inspiration for writing his or her own stories when not using the program. As a bonus one gets to see one's own story shown as an animated film

Installation

Place the Omega-is CD into your CD-ROM drive. In Windows the installation will normally start automatically. Otherwise; find the CD drive on your system (from "This Computer" in Windows) and open it. Find and start the "Omega-is-1-2-Win-install.exe" file. Installation files for Linux and Mac OS-X are also available on the CD-ROM.

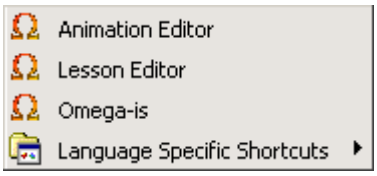
When the installation has started, just follow the instructions on the screen. For instance, you will be able to choose what languages to support.

Note that this is a fairly big installation which will require some time! Count on 5 - 10 minutes to complete the installation.

The program requires 180 MB available space on the hard disk when one language is installed. Every additional language requires approximately an extra 10-15 MB.

The minimum requirement for your computer is the Pentium 3 with 600 MHz and 256 MB RAM. Pentium 4, >800MHz and 512 MB RAM is recommended.

After a successful installation, an Omega-is folder with three new icons (plus an additional folder) will appear on the start menu:



The icons reflect the fact that Omega-is actually consists of three different programs. Two of these (Animation Editor and Lesson Editor) enable the user to edit and customize the program. Changes can be made to existing lessons, or new lessons can be created. Additional information about these features are found in the full manual, and in the help files of the editors (also see our Web pages on www.omega-is.com).

Icon #3 - **Omega-is** - represents the actual program, the one that gives the learner the opportunity to explore the language in an amusing way. And which gives you as a parent or teacher a new educational tool.

Omega-is will ideally pick up the menu language matching the computer system language setting – if available – otherwise the English version will be started. In the 'Language Specific Shortcuts' folder dedicated start-up icons for the supported languages can be found.

Omega-is is currently supporting three user interface languages (English, Swedish and Norwegian), and lesson material in four languages (English, Swedish and Norwegian 'bokmål' and 'nynorsk'). Any installed lesson language may be used regardless of the current user interface language.

Start the program

Quick start:

- Double-click on the Omega icon
- Select ”Guest”
- Select ”Create …” or ”Test”
- Select lesson type and lesson

When starting up the program the user is automatically placed in **learner** or **pupil mode**, which enables the learner to start examining the program, either in the **Create Sentences** mode (the basic activity offered by the program), or in the **Test** mode (testing the skill to interpret animations and to re-produce it in text).
Trying the program as a “Guest” gives the user the freedom of doing so without the results being saved in a specific result file.

Create Sentences enables you to explore language in a fun-filled and error free way. Single words can be explored in the B lessons. Simple two and three word sentences are created and explored in the D and F lessons, and more complex sentences in the H and J lessons. Short stories may be produced in the L-lessons.
In this mode, the user selects words (or short phrases), presented as text buttons, which then are added to a Sentence Line and spoken.
Whatever the user will do (except for quitting) a sentence will be created in some odd way. The skill to do it in the standard way may be gradually developed.
The complete sentence will be spoken again, and an animation corresponding to the meaning of the sentence will be presented.
After the animation the sentence text will be presented again – and **the spoken version and the animation may be repeated once by pressing the Up and Left arrow keys**.
This is the time to discuss the (often more or less crazy) meaning and illustration of the sentence, an excellent and natural opportunity for the teacher to stimulate the learner’s understanding of language, and to encourage conversation and progress.

Test. If, instead, the “Test” option is chosen, the learner will have the opportunity to explore his or her language skills in a game like way. In this case, the animation is shown initially, whereupon the learner is to select words or phrases to create the sentence that best describes what he or she just saw.
When a sentence has been completed a right/wrong response is presented (in optional ways, according to the user settings (see Ch. B and C).

When the user confirms the feedback, a new animation is randomly selected from the sentences/combinations that are available for the selected lesson.
When all combinations have been presented, or when the user chooses to quit, a result summary is presented, and then the user is brought back to the lesson menu.

The above description has presented Test in **Learner** or **pupil mode** where it serves as a complement to the "Create Sentences" activity.

There is also a more structured and ambitious Test option in **Teacher mode**, which contains added "dummy" words, with the purpose of further challenging the learner's reading comprehension (you arrive at the teacher mode/settings by pressing **Ctrl+F2** or **Shift+F2** from the initial menu).

Results (e.g. percentage correct, and response time) of all activities are logged and stored in a result file. This may of course be useful for teachers to view and analyse the results and progress of the learner's activities. Some learners may appreciate the option of being able to challenge themselves by re-taking a test to see whether the score can be improved, or if the score can be maintained in less time (more on test and the result file in ch. C).

Motivation and support

Our experience shows that learners improve if an adult is at hand to give positive reinforcement or support when needed, but also as a conversational partner when the learner wants to discuss his or her experiences. It is important, though, that the initiative stays with the learner as the language material is perused. The learner must be driven by his or her own curiosity and delight (more in on this in chapter G).

Adaptability

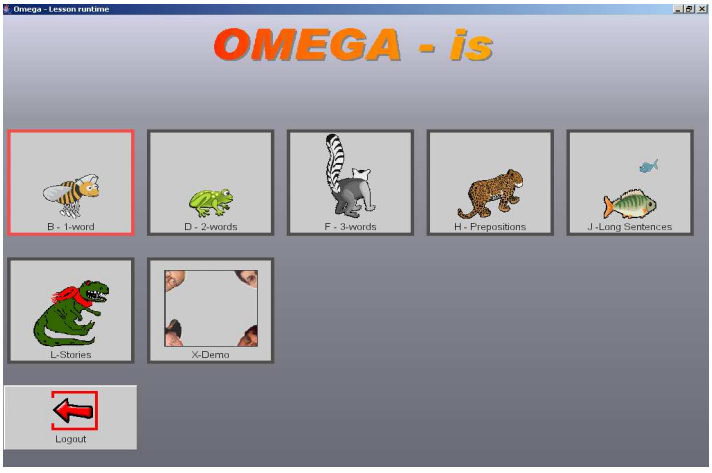
The program can be operated through either mouse or keyboard (for instance the Space and Enter keys) This opens the possibility also to make additional modifications for learners with varied motor abilities who are in need of 1 or 2 switch input.

Chapter B

Chapter B: Create

Creating sentences and events

Omega-is has been created as a multimedia-based computer tool which puts the learner in the driver's seat, enabling him or her to examine language in a playful manner. This is the fundamental concept. The learner has the means to examine everything from single words to phrases (sentences) of different complexity up to short stories. This language material is reproduced as text, sound, and as animations. The learner creates illustrated phrases which stimulate the understanding of language and provide motivation towards further discourse and progress.



Choosing the level of difficulty (B- to L -lessons) is done by clicking on different animal icons

Register a new user (pupil)

To add another learner or make changes to the basic program settings, you need to enter the **teacher mode** by pressing **Ctrl+F2** or **Shift+F2** (see below). In this mode you have access to the program's set-up functions, particular teacher-tests and test results (see further ch. C). To register a new student, choose <New Pupil> (see figure 1 below) and type the learner's name. You will then enter a new dialogue window which enables you to make further adjustments (see figure 2 below).

(1)

(2)

Animation and **Feedback** enables you to alter the speed of the animations. You can also choose to repeat animations and adjust test-exercise feedback (more on this in Ch. C).

Language: This is where you select what language the learner should be working with. Choices here depend on what languages have been installed (available choices on the set-up CD are English (British), Swedish, and two versions of Norwegian.

Look&Feel: Adjustments can be made to individualise button features, text, background and the screen. In addition, you can also adjust size and contrast for optimal personalization. Details concerning keyboard features can also be adjusted: Space for selecting and Enter for executing the choice made.

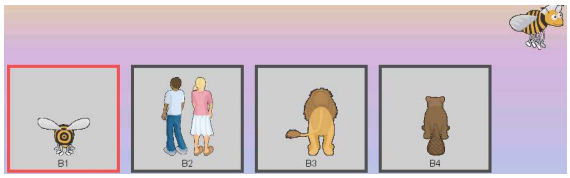
Admin: A picture/photo of the learner can be linked into this feature. This picture will be shown every time the learner starts to work with the program (please note that the picture needs to be stored in the jpeg format!). With "Admin", you can also erase previous learners/pupils and any adjustments (learner data stored in a result file must be erased manually).

After completed registration and adjustments, the learner can start to create his or her own events (sentences). Again use the commando Ctrl+F2 or Shift+F2 to re-enter the **Learner** or **Pupil Mode** and choose **Create sentences**.

Choosing a lesson

The program contains a large number of complete lessons, giving the learner the means to explore language and written text based on his or her own skills and motivation.

B lessons make up the first and least complex level. Singular words are explored, i.e. the learner clicks on the word, then sees the written word appear again, hears the spoken word and finally views the image. To move on to another word, the learner clicks on the text below the image. There are four complete B lessons containing a total of 29 unique words.



The four B-lessons

Words in lesson B1

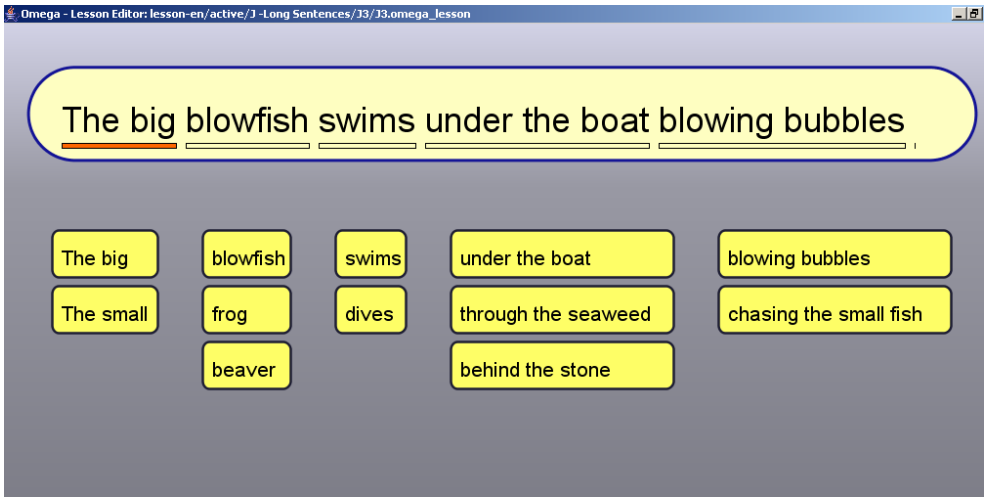
D lessons. Words are put together in pairs – an agent plus an event or object. Examples of events that can be created are ”The girl swings” and “The squirrel dances.” Three lessons are included (D1, D2 and D3), offering the means to create 81 different sentences/events.

F lessons. The F level provides the means to write three-word sentences consisting of an agent/subject, a verb, and an object (s-v-o). The initial phrases contain only two nouns and a verb, thus offering the learner a chance to understand how things work at this level without too many disturbing options. There are a total number of 12 different F lessons with the possibility to create 205 linguistic events.

H lessons. The H level consists of five lessons, where, among other things, prepositions and positional attribution can be practiced. Directional concepts such as “looking out from”, “looking into,” and “jumping out of” are also examined. The genitive appears in some of the H lessons, letting the learner work with phrases with possessive content (such as “the panda’s table”). Sentences are now longer and more complex, putting a considerably higher demand on the learner’s linguistic ability, motivation and working memory. A total of 951 phrases/events can be created at the H level.

J lessons. The J level contains five lessons, including, among other concepts, adjectives and conjunctions. Phrases are now divided into text buttons (see the figure below), providing a number of combination alternatives, and the learner is now approaching a “conventional” manner of writing. The purpose with these lessons is to let the learner practice his or her ability to examine more complex linguistic material, something which, hopefully, will lead up to the creation of

stories (L lessons). Additionally, the learner is given the opportunity to work with sentences where the actors express feelings ("the happy..." and "the frightened..."). J lessons involve increased drilling of the working memory, owing to the increased amount of information that has to be processed by the learner. A total of 699 phrases/events can be created in the different J lessons.



A complex phrase at the J level (lesson J3)

L lessons. On this level, there are two lessons where the learner can create a longer story by combining different phrases. The phrases provide the means to decide who should be the main agent, what he or she should be doing, and in what context the action is taking place. Thus, the learner can create different versions of the same basic story. The purpose is to stimulate the learner's imagination into creating his or her own stories! The story created can be read aloud and/or replayed and printed. It can also be saved to be repeated at a later time.

Language versions

The existing version of Omega-is presents the possibility to work in three different languages: Swedish, Norwegian and British English. These three languages can be installed separately or as a combination. If more than one language is being installed, choose "Language" in the Settings Menu (press **Ctrl+F2** or **Shift+F2** for the *teacher mode*). These options must be set for each learner individually. Thus, it is possible to have a learner work with Omega exercises in both the Swedish and English versions.

Linguistic material

The basic material in lessons B1 through H5 contains 147 unique words. In addition, there are the words included in the L-level stories and the supplemental demo lessons. More specifically the program contains 33 different verbs, 81 nouns,

14 prepositions (for example behind, beside, over, on, under, in front of, through, and in), 12 adjectives (for example blond, redheaded, and small), 2 adverbs (slowly and quickly), one conjunction (and), one definite article (the), one pronoun and two indefinite articles (a/an).

This language material (lessons B to J) makes it possible to create a total number of 1936 events (sentences), L-level and demo lesson material excluded. Many more events can be explored in the story (L-level) and demo lessons.

How to work with Omega-is – an example

Imagine the following scenario: John, a pupil with an autism-spectrum disorder is going to test the program. John has shown an interest in words, and his teacher finds that Omega-is might be a method worth trying. John is estimated to have approximately the language level of a five-to-six-year-old.

To get started:

1. Start the program
2. Press **Ctrl+F2** or **Shift+F2** to enter settings in *teacher mode*.
3. Choose <New pupil> and enter the learner's name. You now have the following options:
    - Animation: Choose whether animations should be repeated, whether a complete phrase should be marked with a sound and an animation speed.
    - Feedback: In test mode, the type of feedback desired can be set for correct/false answers. Feedback can be given as sounds, text, animations or film sequences. The desired frequency of feedback can also be set.
    - Language: Choose language (Swedish, Norwegian or British English) depending on what languages have been installed.
    - Look & feel. The colour scheme can be adjusted, thereby creating a feeling of a unique environment for each pupil.
    - Admin. The teacher can link to a picture of the pupil or alternatively erase the information contained in the settings for a particular student.
4. After settings have been completed, you will re-enter the teacher mode. You may either choose test or go straight to the lessons section. In the case of John, the teacher chose to test at the two-word-level (D lessons). Click on Test and choose pretest 1.
5. John completes pretest 1 for lesson D1. His score is two correct phrases out of nine (22.2% correct, which is logged into his result file, accessible through the Result icon).
6. The time has come for John himself to give the program a try. Pressing **Ctrl/Shift+F2** keys will take you back to the **learner** or **pupil mode**.

- (Please note that there are test options here as well. Tests in learner or pupil mode are chosen at random, while tests in the teacher mode contain a preset number of phrases. Posttests in the teacher mode also contain so-called dummy words, i.e. words that are phonologically or semantically similar to the words tested.)

7. John may now work with the D lessons.
8. When John has worked his way through all of the D lessons and tried the tests in learner mode, the teacher chooses to test John again with the preset tests. By pressing **Ctrl/Shift+F2** you will once again be back in the **teacher mode**.
9. John completes posttest 1 and scores 80% correct. The teacher decides to end the lesson and the program is closed down.

Chapter C

Chapter C: Testing

Why tests?

The test function in Omega-is makes it possible to find out how much a learner understands from the linguistic material in any given lesson. The animations are shown initially, and the learner, by clicking on the text icons, creates a phrase which describes the action. The test can be run in two modes; the **teacher mode** and the **learner mode**. In the teacher mode, the contents are predefined, while in the learner or pupil mode, tests are generated at random. Furthermore, the teacher mode contains both pre- and posttests for measuring and analyzing the effects on language skills and reading comprehension when the linguistic material has been explored by the learner.

Testing in the learner (pupil) mode

In this mode, the learner's ability to express/recreate animated events in text form is challenged in a fun way. By observing the learner's results in these exercises, the teacher can form an opinion about the appropriate level for the learner to work on. The learner-mode tests contain linguistic material identical to the *Create* lessons, and the exercises are chosen from the lessons at random. If the learner solves no more than 20-30% of the exercises, this suggests that the lesson material should be surveyed (see p. 2). This hint comes out of our own experience, but the final decision can only be taken by the teacher/parent who knows the pupil, since learners' needs of security and challenge can vary a great deal.

"Results" from the test-exercises, as well as activities in the Create exercises, are stored and can be analyzed at a later time by a teacher. Read more about the handling of results and about testing in the teacher mode in the section below.

Testing in the teacher mode

To test in the **teacher mode**, press **Ctrl/Shift+F2**. Then click on Test. In the teacher mode there are two pretests and two posttests to choose from. These tests contain more detailed information in regard to the learner's results compared to the test in the learner mode. Furthermore, there are alternative words (dummy words) which are meant to challenge the learner's reading comprehension both

semantically (in regard to content, as in car/bus) and phonologically (in regard to that which is heard, as in car/cart).

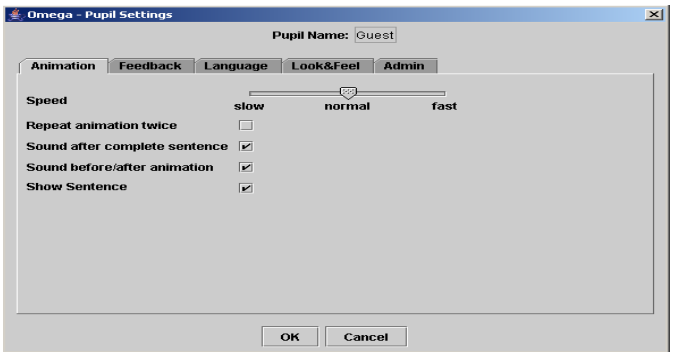
Before starting a teacher-mode test, the learner's name must be entered and the necessary settings must be made (see next section).

How to handle learner/pupil settings

In the so-called **teacher mode** (**Ctrl/Shift+F2**), the program settings, teacher tests and test results are available. In this mode, a new pupil can be added or basic settings of the program be changed (see ch. B). To register a new learner, choose (1) <New pupil> and type in the learner's name. You will then enter a new dialogue box (2).

(1)

(2)



Animation offers the possibility to choose animation speed by clicking on the speed-control bar (slow-normal-fast). Additionally, you may choose whether to have the animation shown twice and whether to have a sound effect when the sentence has been completed, as well as before the animation. Finally, you have the option of having the phrase shown in writing after animation for additional reinforcement of the animation content.

Feedback. When testing reading comprehension, different forms of feedback can be chosen for correct and false answers respectively. Feedback can be given as text, speech, pictures and film clips. You may also choose between pictures and films

within the Omega-is files. Finally, the feedback frequency can be set with the control bar.

Access to personal files is also controlled by the "Set" buttons. This allows for using unique sounds, pictures or film sequences as feedback files. This program feature gives you the option of additional individualization.

Language: Choose the language in which the pupil should work. Choices depend on languages selected during installation (available languages on the Omega-is 1.0 installation CD are English (Br.), Swedish and Norwegian).

Look & feel. Here, you may adjust button, text, background and screen features. This means that every learner may give the program his or her own personalized appearance. Additionally, size and contrast can be adjusted for optimal individualization**.**

Colour theme (Default)

Option 1

Option 2

Option 3

Option 4

Option 5

Option 6

Example: user defined

The pictures above show examples of basic settings or options ("colour themes") that may be selected, and an example of an individual colour theme. There are four different setting options under "change colour". This feature enables the user to adjust the program image at log-in, lesson choice and creation of phrases and stories.

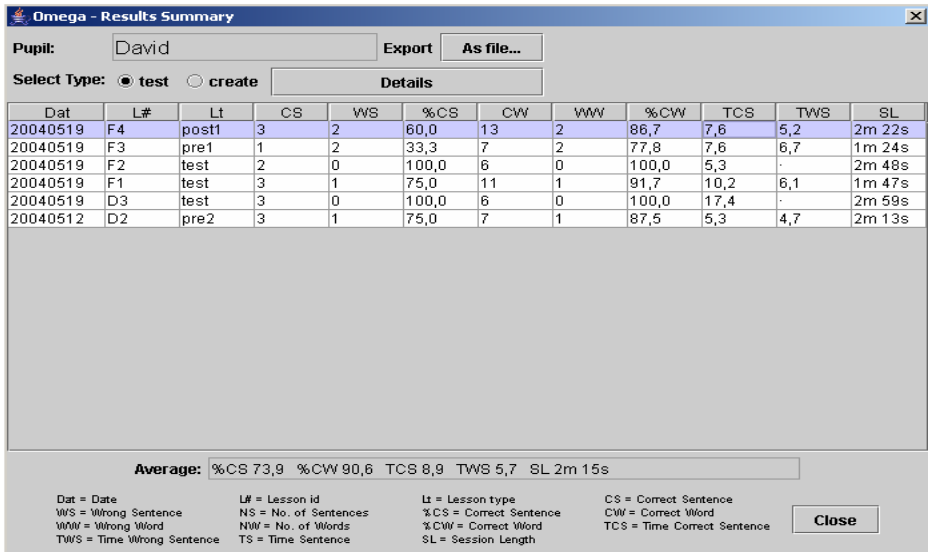
Finally, under Look & feel, the space key function may be adjusted. There are two alternatives: "select next" or "activate selected".

Admin: Here, you may link to a picture of the learner ("set pupil image"). This picture (must be in jpg-format) will then be shown every time the pupil logs in and starts to work with the program. Under Admin, you may also erase a registered learner.

After registration has been completed and any adjustments of the settings have been made, the learner can begin to explore the program. Again use the command **Ctrl/Shift+F2** to return to the **learner** or **pupil mode**.

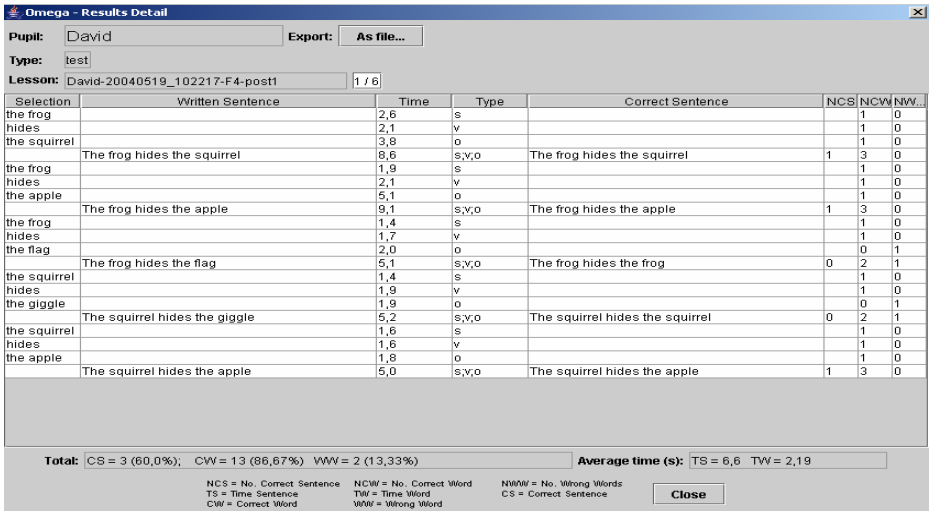
Viewing results

If the adult wishes to study the learner's results, this can be done in the so-called teacher mode: Press **Ctrl/Shift+F2** in the start menu. Then click on the icon named **Result**. Initially, an overview of the tests the learner in question has taken will be shown, including rather extensive information about test types, how much time was spent taking them, answer time, etc (see figure below).



Results: Test-level 1

Should additional information (such as what type of grammatical errors the learner makes) be requested, click on the **Details** button, and level 2 will be displayed (see figure below). In addition to the percentage of correct words and phrases, the exact test phrases and the pupils' answers are shown. We gather from the figure that the learner is completely correct about the first two phrases, but is incorrect about the object in the third phrase. In addition, the answer time is given for every single word.



Results: Test-level 2

Please observe that the result function in Omega-is also offers the possibility to view what the learner has accomplished in the Create mode. Choose "Create" under "Select type" (see picture for Result: Test-level 1).

Analysing Results

By analysing results in detail, the parent or teacher can find out whether the learner is improving his or her reading comprehension and also whether there is a word in any phrase which is easier or more difficult to read. This makes it possible to systematically examine any difficulties and whether these relate to certain word classes being more difficult to the learner. For instance, one study revealed that the object form was the most difficult to children with autism, something which in turn may provide the basis for further drilling or examination (Tjus and Strid, 2000). Teachers also have reported that some learners, though giving answers at a slower pace, have more correct answers, indicating that the children have become more thorough in their work. Other children give their answers both quicker and with a heightened confidence (Tjus et al., 1998). However, the interpretation of the results relies entirely on one's knowledge about how the individual learner functions, and a visual examination will be sufficient in most cases.

For those who wish to do a larger study of the learners' test results, the results can be exported into Excel (or another spread sheet program) or to a statistical program for further analysis. By clicking on the button ***"As file..."** at **"Export"*,** a dialogue box will be opened. Name the file and select where to save it. The file can then be opened/imported as a text file from within Excel or a similar program.

Chapter D

Chapter D: Editing lesson material – the Lesson and Animation editors

Overview

An exercise – ”lesson” – in Omega-is consists on one hand of its contained text material (words/phrases etc.), on the other hand of the media material that is used. The media material consists of recorded speech (for every word/phrase), and of the animations (with their attached image and sound files) or video clips, corresponding to the sentences that can be created in the lesson.

The uniqueness of Omega-is is not only the richness of the contained language material, represented as text, speech and animations. It is also the included tools for editing and complementing this material. These tools are the Lesson and Animation editors.

To manage these tools correctly it is useful to have a somewhat deeper understanding of how the lesson (text) and media material is created and interconnected.

In this chapter we try to give an overview of these issues; How the editors relate to each other, to the lesson and media content, and how this is displayed in the exercises.

In the two following chapters the two editors are described separately and in detail.

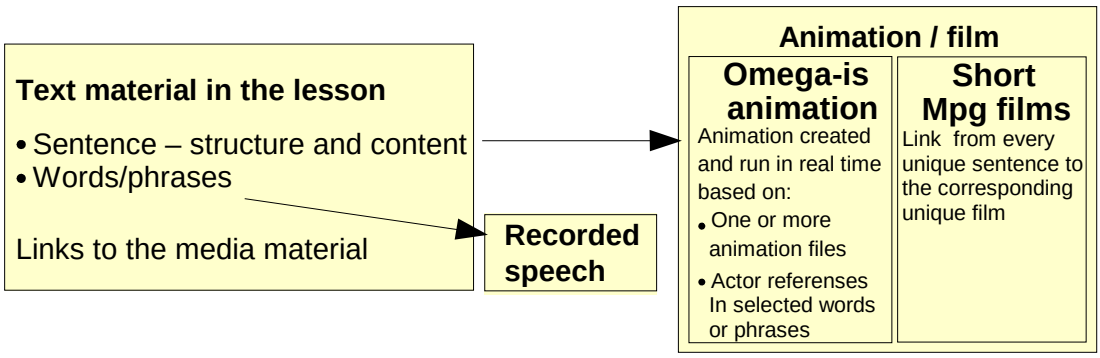
NOTE! We don’t want to spread any illusions about that it will be very easy and quick for each and everyone to create lots of new lesson material in Omega-is. To handle multimedia based content of this kind will in general require quite some media and computer management skills and, in particular, time and interest. To make exercises in Omega-is you will also have to become familiar with how the program and its tools are operating. We who developed Omega-is are not expecting that most teachers or parents will go and grab these tools to produce new exercises. On the other hand we do expect that a limited number of particularly interested persons will want to try and contribute new content. We also

want to show that you may approach the editing tools on quite different levels of ambition. You can start by making small and relatively simple changes and additions to the existing material.

We also want to encourage the initiation of small to more extensive projects – locally, or in co-operation with us producers – to collectively contribute to an increased amount of stimulating and educationally useful exercises for different target users. We encourage all users to watch our web site at **www.omega-is.com** where new exercise content will be made available over time. We will also help to provide adequate education for interested parties. Now, good luck in finding your way to the inner secrets of Omega-is in this and the following chapters, and enjoy!

Exercises – content and structure

Let us start with a figure covering the material that constitutes a lesson :



The building blocks of a lesson : Text, structure, links and media

As shown in the above figure the basic components are relatively easy to distinguish. The text material of the lessons will by now be familiar to the reader, as will the accompanying recorded speech. The distinction between two different media types - **Omega-is animation** and **mpg film** - may be more surprising? This may be because almost all the included lesson content is based on Omega-is animations. An example of the use of mpg films is currently found only in the ”Demo3” lesson in the ”X-Demo” folder – in the sentences ”the cup jumps over the hand” and ”the cup jumps over the cup”. We will below study the clear difference in management of lessons, or individual sentences, based on animation or mpg film respectively – advantages and disadvantages. (**Note!** QuickTime .**mov** film files and .**mp3** sound files may also be used in version 1.2!).

In the above figure, some of the more complex aspects of the lesson material are hidden behind the words ”structure” and ”links”. A couple of lesson examples will show what these structures and links may look like, where the different lesson and media files are stored in the folder hierarchy of Omega-is, and how you give the learner access to new lessons.

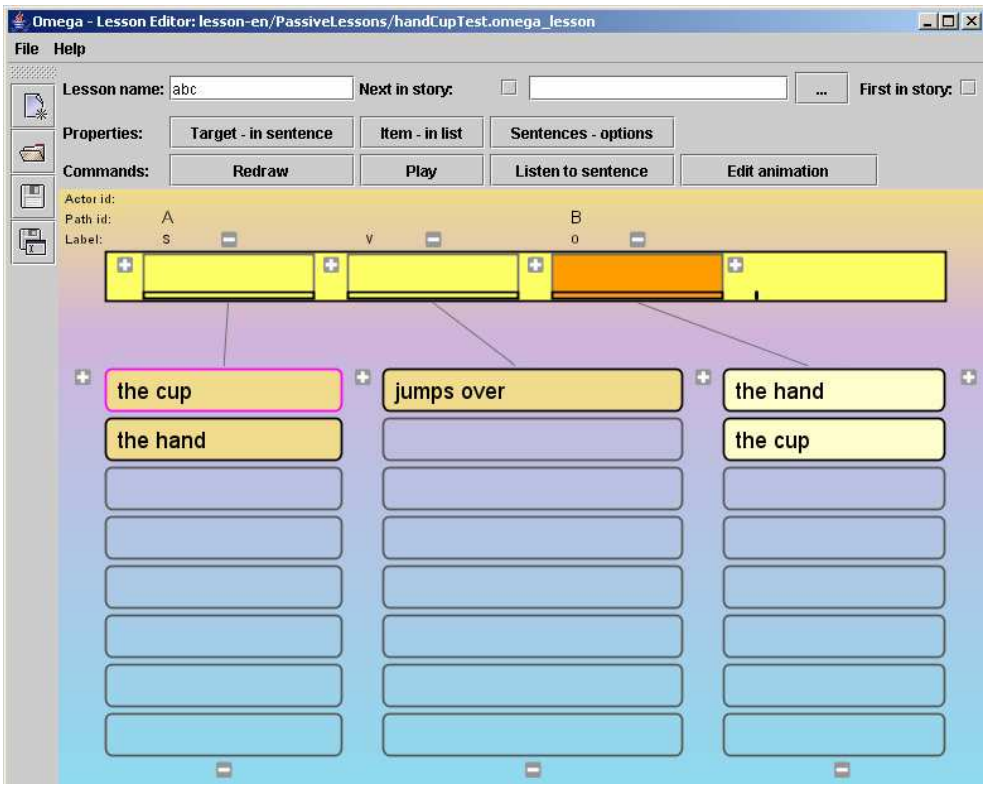
Two lesson examples

We will now study two example lessons: One is based on short video film clips – so called mpg or mov files – of the kind you can generally make with digital cameras. The other lesson example is based on animations made with Omega-is's own animation program. The aim is to obtain a general familiarity with the editing tools, the lessons, and their related media material. The details concerning the use of the Lesson and Animation editors will be covered in the two following chapters.

Example 1: A lesson with video sequences

- Start the "Lesson Editor"
- Select "File" – "Open" - or click the "Open" icon – (the folder)
- The content of the "omega" folder will be displayed: Open the "lesson-en" folder
- Then open the "PassiveLessons" folder
- Open the lesson "HandCupJumpExample.omega_lesson"

The screen should now look like the image below:



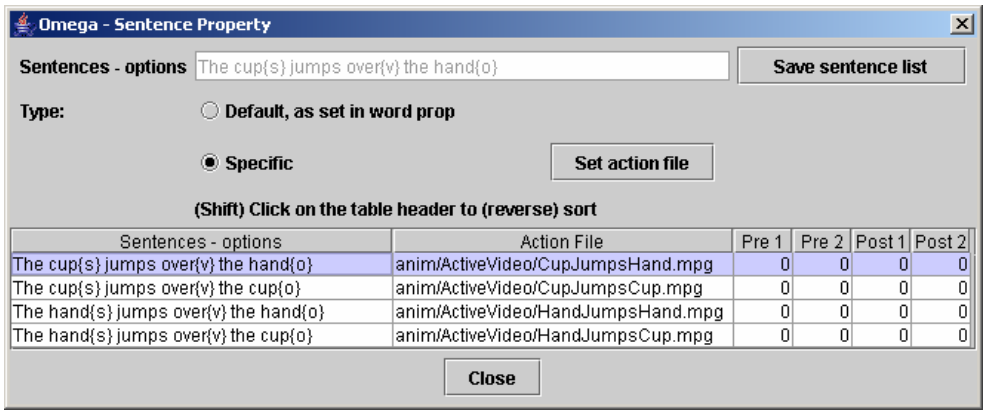
- Click on (e.g.) "the cup" (in the left column), then "jumps over", then "the hand" (in the right column). The words will be spoken and placed in their respective "slots" in the "Sentence Line".
- Click on the "Listen to sentence" button to hear the whole sentence spoken.

- Click on the ”Play” button – and a short video sequence will be presented, illustrating the meaning of the sentence
- Test swapping one (or two) of the words in the sentence by selecting the desired word(s) in the ”word lists”
- Click on ”Play” again to see a video clip for the new sentence.

How is the correct video clip found?

- Click on the ”Sentences – options” button

The following dialogue window will be shown:



Here you can see the four sentences/ which can be created in this exercise. As you can see, every sentence has been linked up to a ”Specific” so called ”Action File”, i.e. short video clips in the form of MPG or MOV films. Note where the films are stored – the ”relative path”; ”anim/ActiveVideo/…”. This path has its root in the ”omega” folder, i.e. the folder where Omega-is is installed on the hard disk of the computer. The bottom sentence in the list – ”The cup jumps over the hand” – is illustrated by the film ”CupJumpsHand.mpg”.

- Try to (re-)find this film by:
  1. Clicking on / selecting the bottom sentence in the list
  2. Clicking on the ”Set action file”
  3. Finding the file ”CupJumpsHand.mpg” by browsing – from the “omega” folder – to the ”anim” folder – and then further to the ”ActiveVideo” folder. Select ”Cancel” when you’ve found the file.

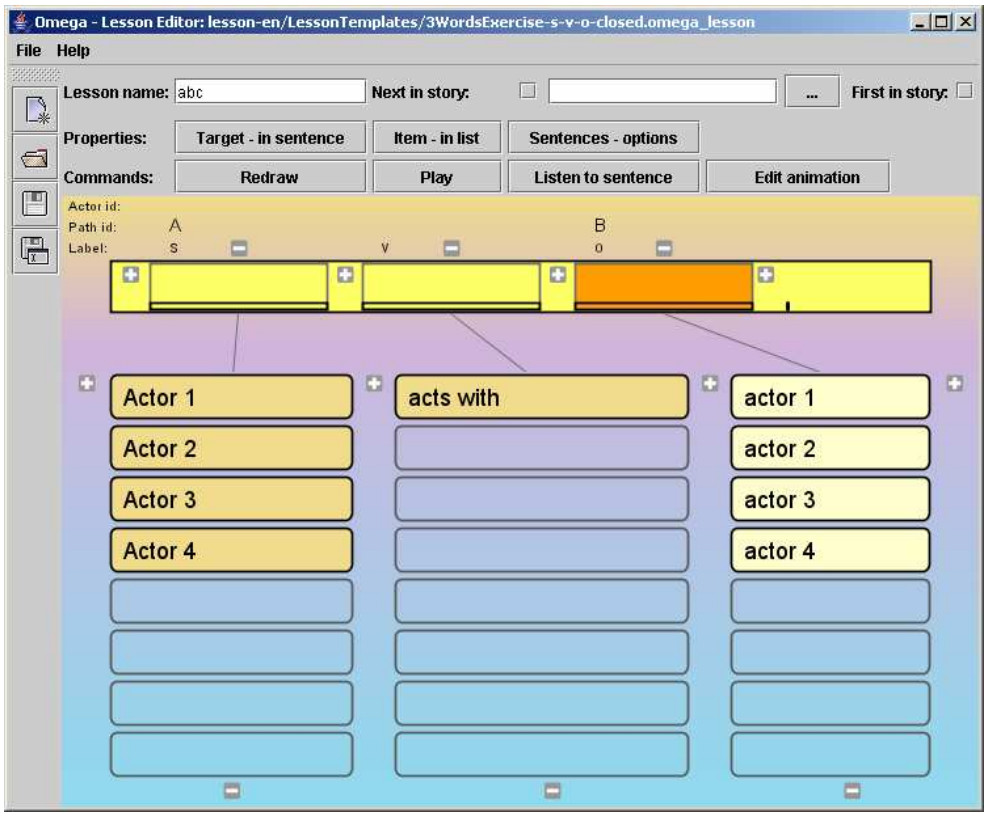
This example has shown how a simple lesson based on video clips can be created:

- Two nouns (”the cup” and ”the hand”), which can act as a subject or an object, plus a verb phrase (”jumps over”), can together be combined into 4 sentences.
- These sentences require 4 unique illustration films – here stored in the folder ”../omega/**anim/ActiveVideo**/”.
- The lesson ”HandCupJumpExample.omega_lesson” is English, and will therefore be found somewhere in ”…/omega/**lesson-sv**/…” – here in the ”PassiveLessons” folder.
- The lesson is also referring to audio files. More on this in the next example.

Example 2: A lesson with Omega animations

- Start the ”Lesson Editor” – if it’s not already running
- Select ”File” – ”Open” - or click on the ”Open” icon – (the folder)
- Open the ”lesson-en” folder
- Then open the ” LessonTemplates” folder
- Open the lesson ” 3WordsExercise-s-v-o-closed.omega_lesson”

The screen should now look something like below:



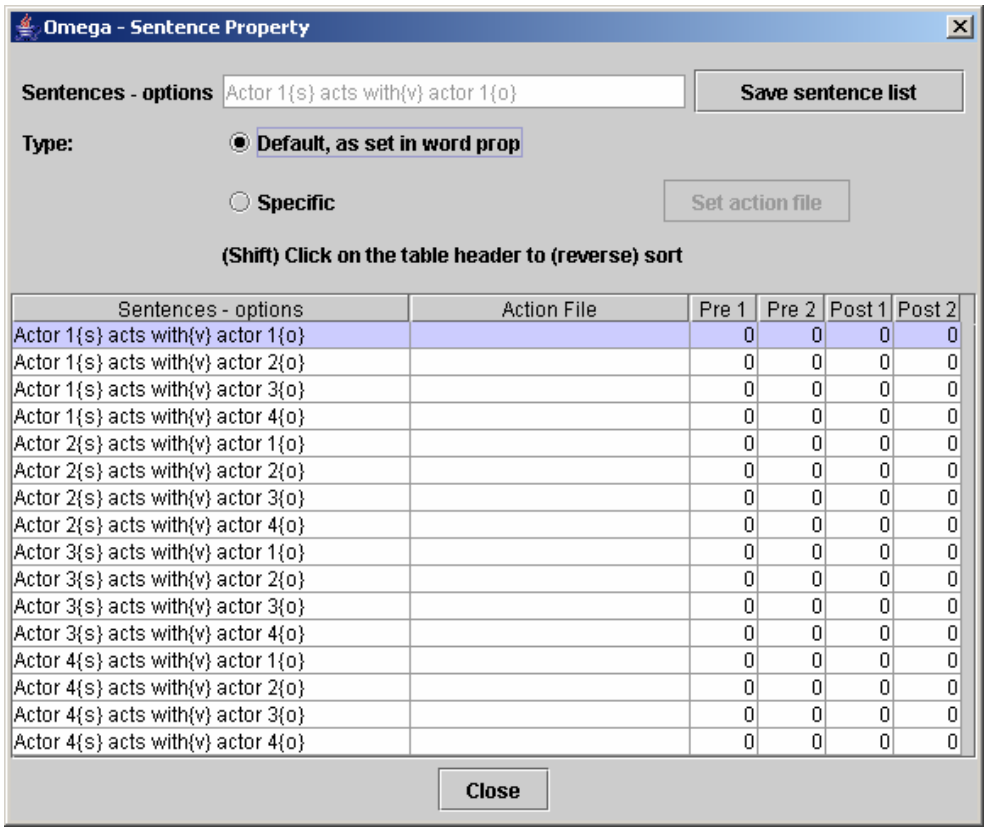
As you may see, this is the same lesson template which is displayed when the Lesson Editor is started.

- As in the previous example, you could start by creating a few sentences by selecting words in the lists. Use the ”Listen to sentence” and ”Play” buttons to examine the results!

In this lesson, no video sequence is presented on ”Play”. Instead an animation is run, created in Omega-is’ own animation program – in different variations, depending on which words – and thereby ”actors” – that have been selected for the sentence.

In this lesson 16 different sentences can be created (4 x 1 x 4 = 16). If video clips or traditional animations would have been used, 16 different films would have been required to illustrate all combinations. But here we can make do with just one (1) animation, which is created and performed with a cast of actors – finally decided by each unique sentence.

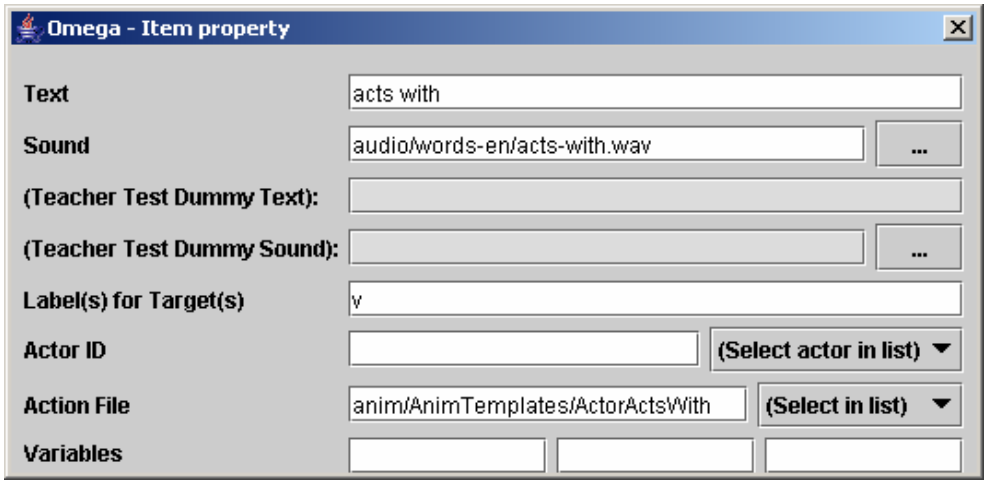
- Click on the ”Sentences – options” button to inspect the overview of possible sentences! Compare to the previous example:



Now you see no links to an MPG film for each sentence. Instead the ”Default, as set in word prop” alternative is ticked.

So where can we find these word properties with links to an animations and actors?

- Close the ” Sentences – options” dialogue (above)
- Click on the ”Item - in list” button (or right click on any word in the lists). An ”Item property” dialogue is opened.
- Select the button with verb phrase ”acts with” in the list. The following dialogue should be displayed:



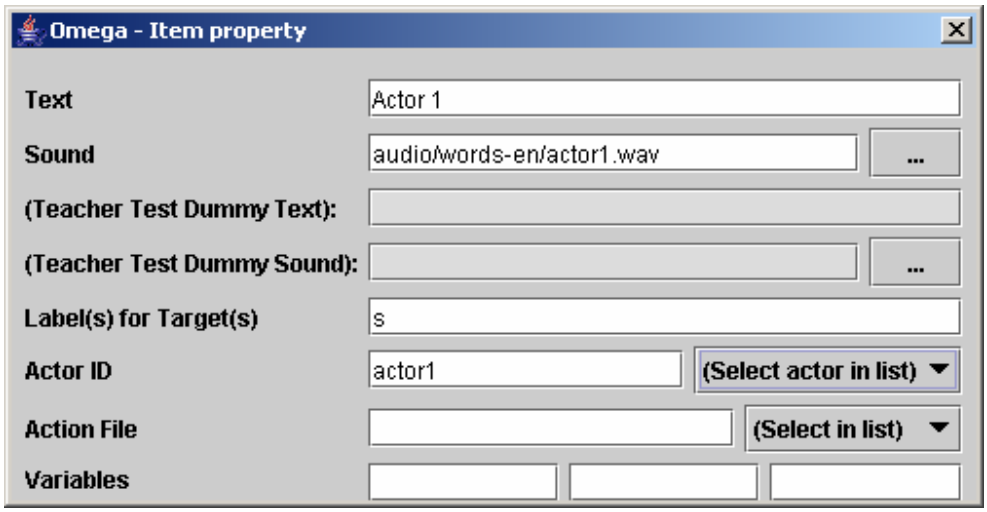
Let's take a closer look at the content of the "acts with" item. For the moment we'll concentrate only on the two entries referring to media files:

- In the "**Action file**" entry there is a reference to a file called "anim/AnimTemplates/ActorActsWith". Click on the "Select in list" button and browse for the animation file "**ActorActsWith.omega_anim**". It will be found in the "**AnimTemplates**" folder which, in its turn, is found in the "**anim**" folder in the "**omega**" folder. Select "Cancel" when you've found your way to the file.
- The second reference to a media file is found in the "**Sound**" entry. It links to the sound file "audio/words-en/**acts-with.wav**" (or **acts-with.mp3**). This is an audio file with recorded English speech, placed in the "**words-en**" folder, which is found in the "**audio**" folder – within the "**media**" folder – in the "**omega**"-folder.

Does it feel overwhelming to keep track of all these documents and folders within folders? That's fully understandable. This will become clearer after a while, and we'll support you with an overview of the folder and document structure in the next section.

The names of the media files are generally in English – for all languages – to make it as easy as possible to manage links to corresponding media and lesson files for different language versions of Omega-is.

- Now click – with the Omega - Item property dialogue still open – on one of the other items/words, e.g. "actor 1" in the left list.



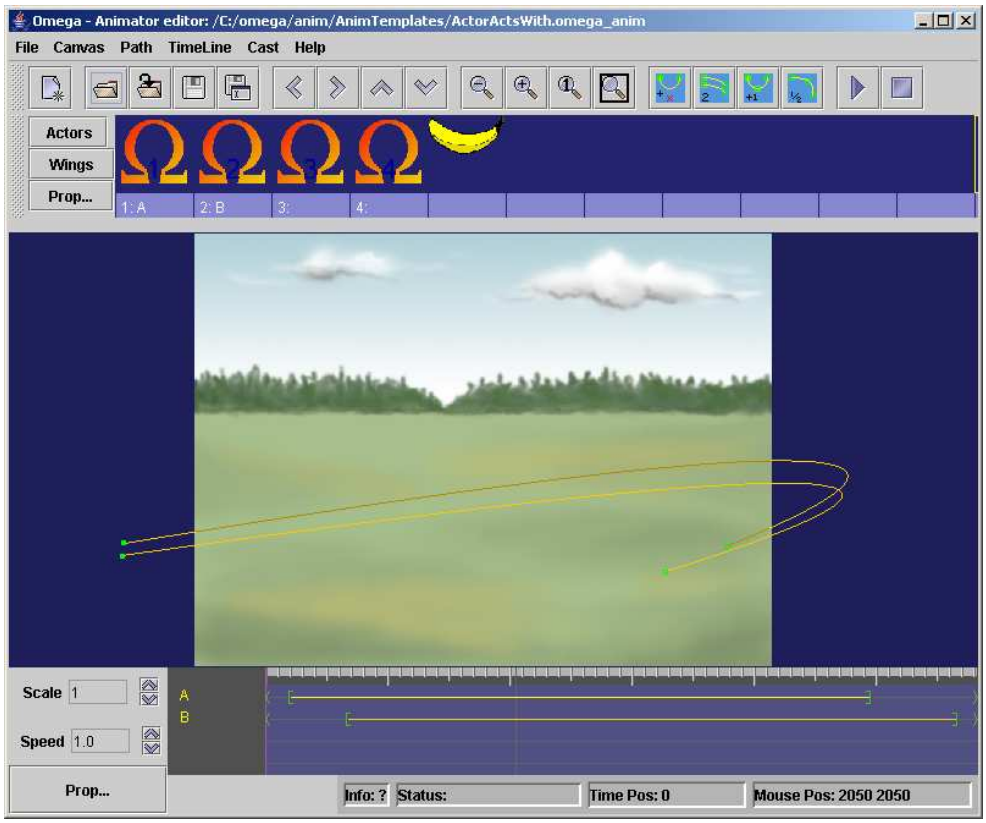
The content of thei dialogue now shows the properties for this item/button: The **Text** and the link to the appropriate **Sound** (like before). Note that for this item there is no link to any **Action file**, but instead there is an **Actor ID** – "**actor1**".

- Click on the "**(Select actor in list)**" button. There you should find the following "actors" to choose from: **actor1**, **actor2**, **actor3**, **actor4** and **banana**. These actors are available in the Omega animation "**ActorActsWith.omega_anim**" – which was linked to in the properties of the previous item ("acts with").

Further details concerning the item properties will be left until later.

We'll approach the end of this example by making a first acquaintance with the **Animation Editor**:

- Make sure you have created a full sentence in the Sentence line
- Then click on the "**Edit animation**" button. The Animation Editor will now be started – with the animation which is linked to the sentence:



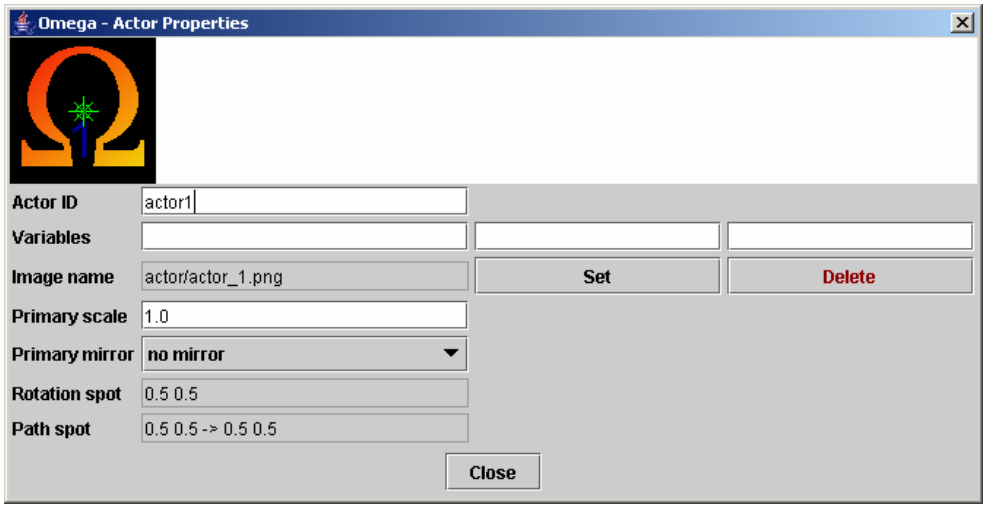
Here we can see the animation "**ActorActsWith.omega_anim**", ready for editing. The four actors that were listed earlier in the Lesson Editor can be found in the Actor List in the upper part of the screen – four Omega actors. In addition there is an unused banana actor.

We will not now go very much deeper in the exploration of the secrets of the Animation Editor; We will return to this later in Chapter F. For now we will limit ourselves to taking a brief look at the Actor Properties:

- Open the Actor Properties by right clicking on one of the actors, e.g. the first Omega actor, to the left in the list. The dialogue window (below) shows the properties of the currently selected actor.

Notice the Actor ID – e.g. "**actor1**" – which is called upon from the lesson in order to link each "actor word" in the lesson to the corresponding actor in the animation!

Also note that this is where the appearance of each actor is decided by the link to a certain image file, in this case the image ”**actor_1.png**” in the ”**actor**” folder – which is found within the ”**media**” folder.



Now close the Actor Property dialogue. Please note that the two first/leftmost actors, which have the ”labels” 1 and 2, are linked to their respective animation path via a ”Path ID” (”A” and ”B”). The paths are represented at the bottom of the screen, along the ”Timeline” – with their respective IDs – and are also found on the animation ”stage”. (These ”Path IDs” - ”A” and ”B” – are also referred to in the Lesson Editor – on top of the target word boxes in the Sentence Line. More on this in the chapters on the Animation and Lesson editors below.)

Feel free to test run the animation in the Animation Editor by clicking on the ”play” button in the top right corner. The two actors with the labels 1 and 2 will be used, which are also linked to the paths IDs ”A ”and ”B”. You can test another actor by dragging one of these labels to the position under the desired actor.

Finish this example by closing the Animation Editor (confirm with a Yes), and after that also the Lesson Editor.

The files and folders of Omega-is

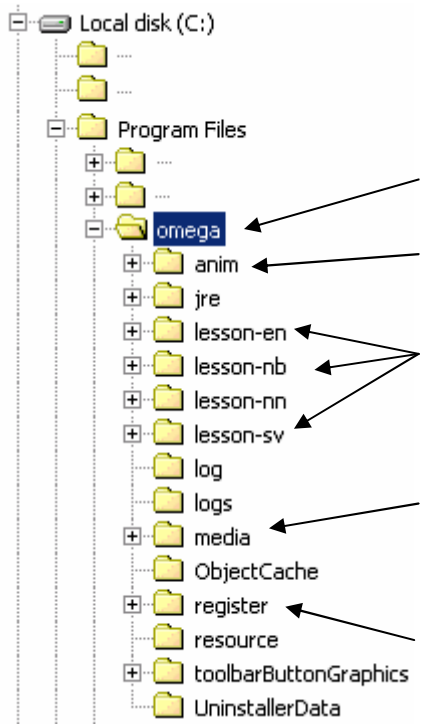
The above examples have given a first insight in which documents/files that are used for the lessons and animations of Omega-is. It should already be obvious that there are a lot of them, and we have still just scratched on the surface. This is not said to scare you off, but to give a realistic understanding of the complexity that is unfortunately more or less necessary for a multimedia program of this kind.

You don’t need to have a full control over all this to start playing around a little with simple editing though. The program will give you support to find your way. But more insight will make it easier to understand and avoid unnecessary problems.

To support your navigation in this territory, we provide you with an overview of the folder structure of the Omega-is installation below, including some information about where different files are stored in this folder structure.

The installation folder of Omega-is – an overview

In Windows Omega-is is normally installed on the hard disk in an "**omega**" folder under "**Program Files**" folder/directory ("Program" in Swedish and some other systems). The basic directory structure should look like this:



The Omega-is install folder

The "**anim**" folder contains the animation files and video clips that are used to illustrate the language material of Omega-is.

The "**lesson**" folders contain the lesson files – for the installed languages. The English ones are found in "**lesson-en**", the Swedish ones in "**lesson-sv**", etc.

The "**media**" folder contains all the media material of Omega-is (except the video clips in "anim"), i.e. all image and sound files (including those used in the animations), recorded speech for the installed languages, and more.

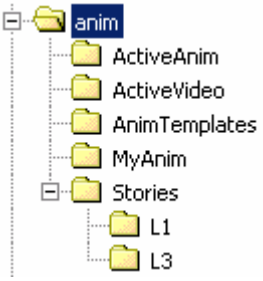
The "**register**" folder contains settings and data for the students/pupils that have been set up.

The other folders are skipped, as they are of a more technical character.

So let's proceed by taking a closer look at the folders highlighted above.

The "anim" folder:

The animation files of Omega-is are text documents in the XML format. They contain no media material in themselves, but refer to images and sound files stored in the "media" folder. The animations are organised in the following folders:



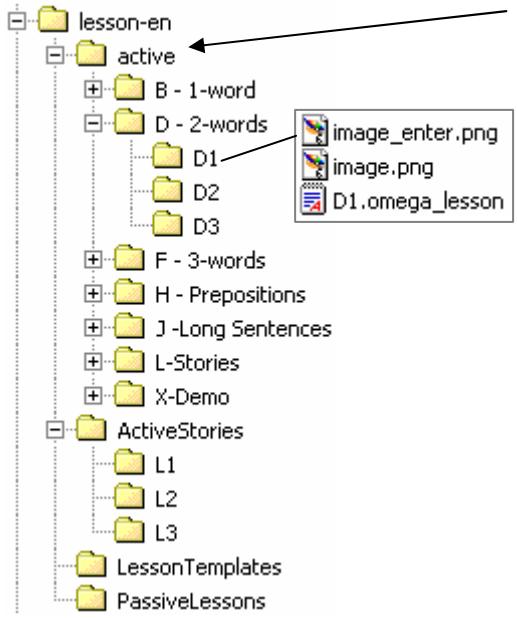
"**ActiveAnim**" – contains all animations that are in use, except those belonging to the stories (L1 and L3) which are found under "**Stories**".

A few animation templates are found in "**AnimTemplates**.

An empty "**MyAnim**" folder is prepared for home made animations.

The very limited number of video illustrations in Omega-is are stored in the "**ActiveVideo**" folder.

The ”lesson-en” folder - English lessons:



The ”**active**” folder contains the lessons that are available in the pupil menu – the B, D, F, H, J, L and X lessons, etc.

The program itself creates the menus according to the content of the ”**active**” folder. This can in its turn contain folders, and finally a folder with an ”**xxx..omega_lesson**” file which launches a lesson – such as the ”**D1.omega_lesson**” in the example to the left! The lesson files are (like the animation files) stored in XML format.

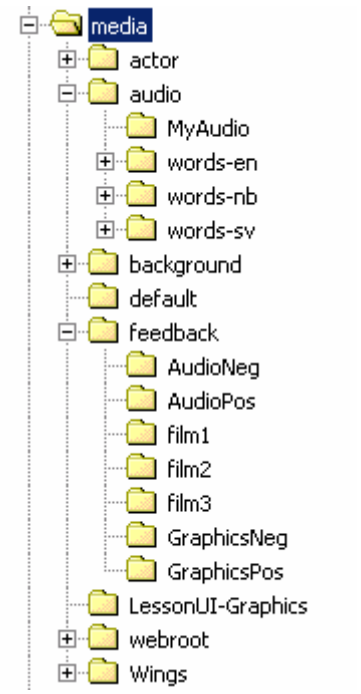
Each folder must contain an image called ”**image.png**”, and ideally also another one called ”**image_enter.png**”. These images represent the folders or lessons in the menus.

The ”stories” material is placed in a special ”**ActiveStories**” folder – except the first part of each story, which is found in the L folder under ”**active**”.

Apart from this there is the ”**LessonTemplates**” folder, and the ”**PassiveLessons**” folder for work material etc.

The ”media” folder:

The ”**media**” folder contains a lot! We will have to limit ourselves here to giving a brief overview of all this content:



The ”**actor**” folder contains all the image files that are used for actors in the Omega animations. They are distributed over a number of sub folders. Some image files will be duplicates, or variants, found in more than one folder. The image file format is PNG (Portable Network Graphics) with transparent background (either 256 colours + indexed background, or millions of colours + so called Alfa Channel). There is also a folder for your own new actors – ”MyActors”.

The ”**audio**” folder contains most of the sound files of the program. They are all in so called wave or mp3 format (”xxx..wav” or ”xxx..mp3”), and they have two kinds of functions:

1. The sound effects which are used in the animations – stored directly under ”audio”
2. The recorded speech files for the lessons – stored within folders for the installed languages – ”words-en”, ”words-sv” etc.

There is also a folder prepared for your own sounds – ”MyAudio”.

The ”**background**” folder contains the background images for the animations – usually in the JPG format. Here is also a folder for your own material - ”MyBackgrounds”.

In the ”**feedback**” folder there are films, images and sounds used for feedback during runtime - in the ”Test” mode.

In the ”**Wings**” folder you will find the images used for “props”, “set pieces or “side-scenes” in the sceneries of Omega animations.

The ”register” folder – pupil settings and data

Here you will find a folder for each registered pupil. The folders contain one file for the pupil settings – ”pupil_settings.xml”, and possibly one pupil image (”id.jpg”). These files are created when a new pupil is entered, and will be updated when the pupil settings are altered – that is in ”Teacher” mode.

Additionally you will find date stamped recordings of the pupil activities, which can be further studied under ”Results” in ”Teacher” mode.

Summary - conclusions

This brings us to the end of this introduction to the editing tools of Omega-is, and its data files and directory structure.

We finish off by re-iterating the two lesson examples described above, and by drawing some conclusions about the two types of Omega-is lessons which they represent:

1. A lesson based on video sequences

Features: Is based on links to recorded **video clips** for the sentences, and on recorded **sounds** for the individual words/phrases

Pros: **Simple structure** - fewer references - just video/mpg/mov files (for sentences) and sound/wav/mp3 files (for words/phrases) to keep track of.

No need to master the Animation Editor to produce your own material.

The dramatising and filming of scenes, representing language expressions, could become an inspiring part of an educational strategy where Omega-is is used to actively perform and process concepts, relations, situations and feelings and their corresponding language representations.

Cons: Requires one film per possible sentence, which could mean an awful lot of space consuming media material for more complex exercises.

Many sentences/scenes could be difficult to represent by simple video clips (unless starting up more ambitious projects with animated film or film tricks of course).

2. Lessons based on Omega animations

Features: Are based on links to **Omega animations** for the sentences – with **references** from contained words/phrases to different actors in the animations, and on recorded **sounds** for the words/phrases.

Pros: **Effective** and **flexible**: One animation can represent several (sometimes all) sentences of a lesson. A lesson, containing hundreds of possible sentences, can be built on just a few animations.

Cons: **More complex structure** – several kinds of references (and possible variables) to keep track of - from the words in the lesson to different animations, and to different actors and their paths in the animations.

You will of course have to learn to handle the Animation Editor to make and edit material based on Omega animations.

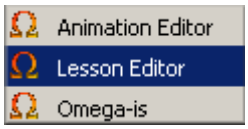
Chapter E

Chapter E: The Lesson Editor

Overview

The Lesson Editor allows you to create new lesson material. This could range from minor adjustments of existing material, to the creation of brand new lessons and lesson content to meet the needs and motivational factors of a specific student

The Lesson Editor is a self-contained program which is opened by selecting **Lesson Editor** from the Start Menu.



NOTE! The Lesson Editor records whether a lesson has been changed! If changes haven't been saved, this is marked in the "title line" of the window, and a warning is given on "Quit", "Open" or "New" commands.

Tweaking existing lessons

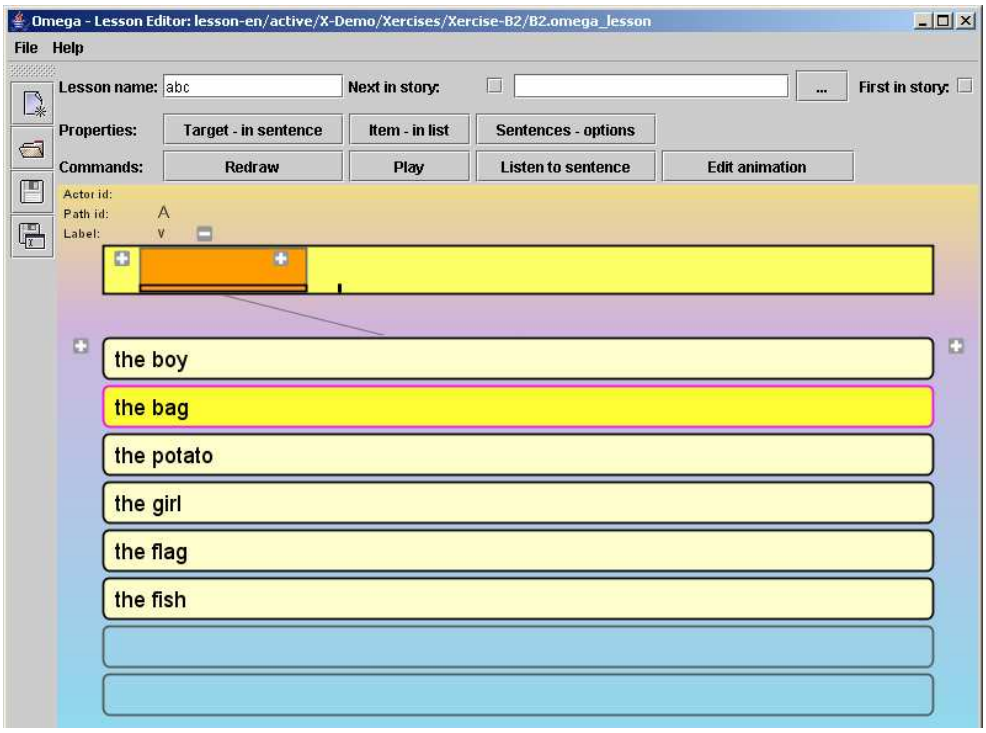
When started, the program presents a default option allowing you to create a new lesson based on a lesson template (see below). Now, however, the task is to make changes in an existing lesson.

1. Click on the Open symbol, or select File – Open.
2. Open the language folder containing your existing lesson e.g. lesson-en for an English lesson.
3. Open the Active folder.
4. Open the folder representing the lesson level (B-D-F etc.) for your target lesson
5. Then find and open the actual folder of the lesson you want to tweak – and finally open the lesson file itself (with a name ending with ".omega_lesson").

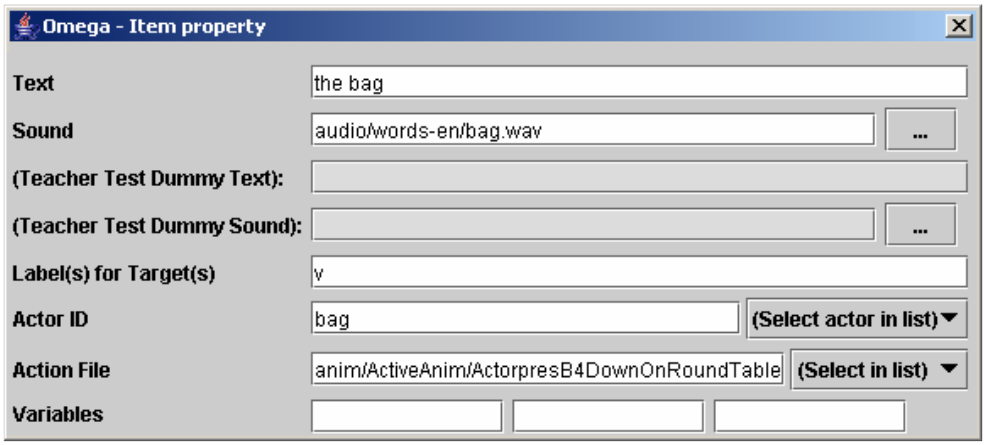
Example 1: Remove a word/item from an existing lesson:

Let's say you want to remove a word that is so disturbing for your student he/she has problems to use the lesson. The word, and its links to animations and sounds will be removed. In the following example the word "the bag" in the Xercise-B2 lesson will be removed. Do as follows:

- Start the Lesson Editor and the lesson Xercise-B2 in the " Xercises" folder in the "X-Demo"-lessons. If a word would really have to be removed, the procedure would have to be repeated for each lesson containing the word.



- When the lesson is opened, all the contained words are shown in the items list.
- Click on the "Item - in list" button – or right click on any of the words in the items list. Then the "Item: Property" dialogue is displayed (see below).
- Then click on the item "the bag" in the list. The properties for this item are shown in the dialogue window, as seen below. Place the cursor and erase the text of the field "Text". Close the dialogue and save the changes (before the Lesson is closed) – by clicking on the disk icon (Save) or open the File menu and select Save. Now the item "the bag" will not be there when working with this lesson.



Example 2: Add a word/item to an existing lesson

- Open the lesson to which you want to add a word. Again we will use the lesson Xercise-B2 as an example. Browse via the ”lesson-en”, ”active”, ”X-Demo” and ” Xercises” folders to find the lesson. Open it so that the list of words/items of the lesson is displayed.

- Click on the ”Item - in list” button – or right click on one of the items in the list – so that the ”Item property” dialogue is opened.

- Then click on an empty item button – below the current words of the list. It is highlighted with a coloured border. Place the input cursor in the “Text” field of the dialogue and type the item you want to add, e.g. ”the vacuum”. The letters will appear in the item button. (This item – ”the vacuum” – happens to be represented by an actor in one of the animations. If you want to add a new item that cannot be illustrated by an actor in the target animation, then you will have to start by adding a suitable actor (represented by a corresponding image file) to an existing or new animation – see the next chapter F.)

- Click on the browse button ”…” next to the **Sound** field. Find the sound file for ”the vacuum” recording (vacuum.wav/mp3) in the Media/Audio/words-en folder). Double click on the file, or select it and then select the ”Load” button. This completes the linking to the recorded sound file for the item.

- New sound files can be recorded via a microphone connected to the microphone input of the computer. You can use the ”Sound Recorder” utility of Windows, but we can for example recommend the free program **Audacity** (http://audacity.sourceforge.net), the sound utility of the CD/DVD package Nero, or some of the many other utilities for sound editing that can be found in the market (e.g. GoldWave, SoundForge etc.). NOTE! The sound quality of files recorded directly into computers may vary a lot. It may be a good idea to consult a person who knows a little about sound recordings to improve the quality. Save sound files in the **wav** or **mp3** format, and put them in the ”media/audio/words-en” folder (or appropriate for other languages).

- To establish a link to a representative animation for the item, first an animation - or so called ”Action file” – must be chosen, which contains an actor that represents the item. This is done by browsing to the directory for animation files: Click on the ”Select in list” button - to the right of the Action file field – and then on <Select file …>. In this case, find and select the animation file ”B1.omega_anim” in the ”ActiveAnim” folder – and then click on “Select”.

- Finally an Actor-ID must be entered, i.e. the unique label that has been assigned to the actor representing the item in the animation. First click on the item button for ”the vacuum” so that the item text is entered into the ”Sentence line”. Then click on the ”Select actor in list” button (to the right of the Actor-ID field). There you should be able to select the actor ”VacuumCleaner”. (You could also type in this Actor-ID.) Test run!

- Close the dialogue(s) and Save the changes before closing the Lesson Editor.

Create new lessons

Existing material can be used in novel combinations to create new lessons. A convenient way to do this is to copy a suitable lesson folder in the "active" folder of the language used, e.g. the D2 folder in lesson-en. The copied folder could be named D4 (the installed version contains D1-D3). Then open this copy D4 in the Lesson Editor. New words/items are entered in the way described above. Appropriate action files (animations), actors and sound files are linked in etc.

If a new lesson with completely new content is created, then corresponding new sound recordings must also be provided, as well as animations with background, wings and actor images, alternatively video clips, in the dedicated folders. The next Animation chapter will tell you more about this. For one, two and three item lessons there are some templates in the Lesson Templates folder.

Below are two examples:

1. How to create a new three item lesson, based on an existing lesson, and on existing animations and media material
2. How to create a new lesson with completely new material based on new video clips and sound recordings

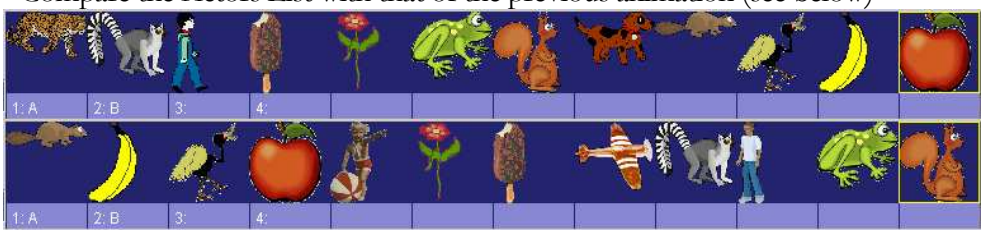
A new three item lesson – existing material

- - Start the Lesson editor – select File - Open
  - Find the "lesson-en\active\X-Demo\Xercises\Xercise-Demo1" folder
  - Open the "demo.omega_lesson" lesson found there

- This is a lesson based on the activities "chases" and "jumps over". The lesson utilises two animations to illustrate these actions.
  We will now re-use this material with alternative actors, to be able to play with words and sentences that are not supported in the original set of exercises.

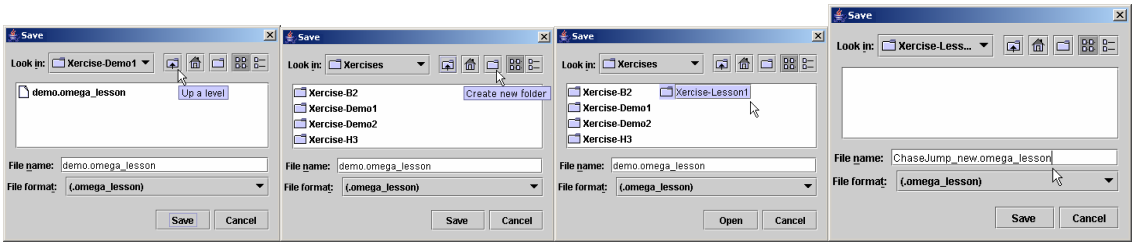
- - Click on the word "chases" so that it is entered into the Sentence Line.
  - Then click on the "Edit animation" button – the Animation Editor is opened with the animation linked to "chases" (ChaseChased.omega_anim)
  - Note which actors that are available in the Actors List of the animation!

- Close the Animation Editor – then click on "jumps over" (which replaces "chases" in the Sentence Line) and re-select the "Edit animation" button – the Animation Editor is re-opened with the "JumpOver.omega_anim" animation
  - Compare the Actors List with that of the previous animation (see below)

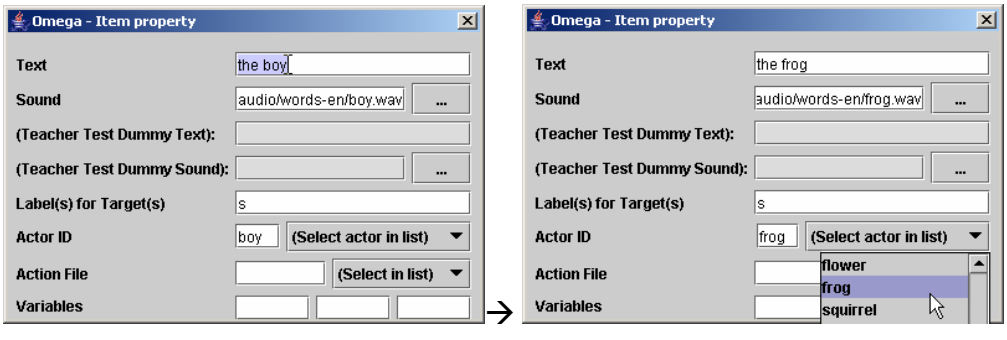


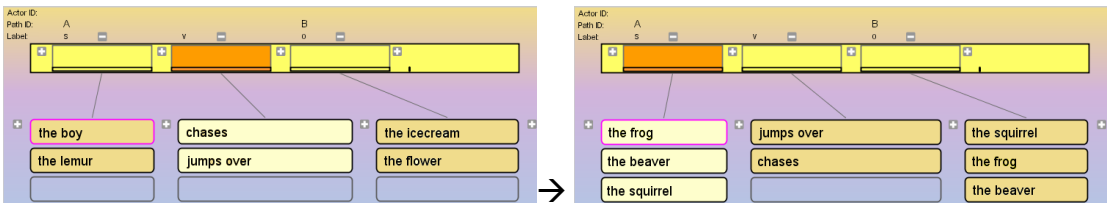
Apparently, among others, a beaver, a frog and a squirrel are found in both.

- We therefore choose to create a "chase"/"jump over" lesson with "the frog", "the beaver" and "the squirrel" actors (The above animations could of course be saved under new names, and new common Actors Lists for both these animations could be edited and used for the new lesson):
  - Select "File" – "Save as" to save the open lesson under a new name. A Save dialogue will be displayed according to below.
  - Move one level up in the directory structure:
  - There (in the " Xercises" folder) create a new folder, click once and then a second time on the folder name (New folder). Name it "Xercise-Lesson1", enter that folder and save the file as e.g. "ChaseJump_new.omega_lesson".



- Now it's time to change the content of the lesson:
  - Produce a sentence in the Sentence Line (e.g. "The boy chases the icecream")
  - Open the Item property dialogue (click the "Item - in list" button or right click on an item) and then click on "the boy" to view the properties of that item.
  - In the form, replace the text "the boy" by "the frog"
  - Click the "…" button next to Sound and locate the sound file "frog.mp3" or "frog.wav" in the "audio\words-sv" folder.
  - In the Actor ID field, erase "boy" and replace by selecting "frog" from the "Select actor in list" list box (or by typing it in).
  - Continue by replacing/entering data for all the three new actors; the frog (frog), the beaver (beaver) and the squirrel (squirrel) in column 1 and 3 according to the images below.





- Test run, adjust and save the new lesson – test in lesson mode.

A new three item lesson – new material

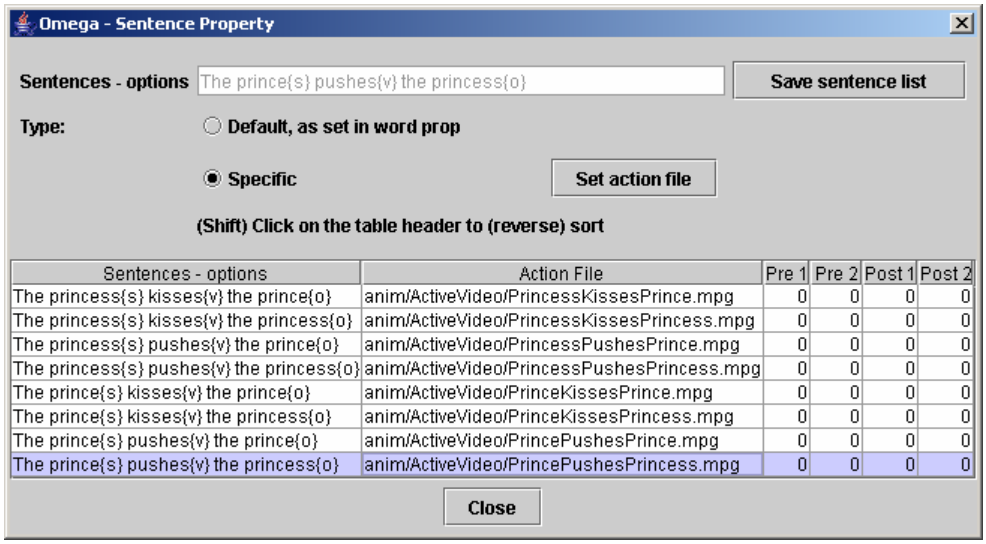
We will now see how a new lesson can be created based on brand new media material. We'll choose to build on **video clips**, as this will not require any knowledge about how to handle the Animation Editor and all image and sound file management needed for the creation of animations (see chapter F).
We will need a number of short video clips that illustrate the sentences we want to play with. We will also need sound recordings for the contained words or phrases

- We start by planning the content of the lesson, resulting in the following:

the princess	kisses	the prince
the prince	pushes	the princess

  This will require 2×2×2=8 video sequences to cover all possible sentence combinations, and 4 sound recordings for the contained words

- Now it's time to make sure that all these media files are put in place – either by recording them ourselves, or by relying on some other resources.
  - Record the four words – see the instructions in the next section – and place the sound files in the "media\audio\words-en\MyWords" folder under the omega folder. We suggest all files, for all language versions, are given English names – to simplify the maintenance of multiple language versions – i.e.: "princess.wav/mp3", "prince.wav/mp3", "kisses.wav/mp3", and "pushes.wav/mp3". (Why not support a second language right away? Place the files in a corresponding folder within "words-xx".)
  - Now dramatise the 8 actions – either with live actors, or using hand dolls or similar means.
  - Record short clips using the video recording option of a standard digital camera, or a real video camera.
  - Transfer the video files to the computer and place them in the "anim\ActiveVideo" folder, under the omega folder.
    NOTE! The video clips should be very short, around 3 – 10 seconds. NOTE also that video files must be of the MPG or MOV type! Some AVI files may be used after being renamed with the ".mpg" file suffix.
- Start the Lesson Editor (if it's not already running)
  - Open the lesson "3WordsExercise-s-v-o-VideoClip" in the "lesson-en\LessonTemplates" folder.
  - Select "File"- "Save as" and save the lesson in a new folder e.g. within the "lesson-en\active\X-Demo\Xercises" folder.
  - Open the property dialogue for "Item - in list" and enter the words according to the lesson layout above. Link up the newly recorded sound files for the items (found in the "media\audio\words-en\MyWords") folder.
    NOTE! In this lesson, based on video clips, no Actor IDs are needed, neither any so called action file need to be linked in the property dialogue.
  - Close the item property dialogue when finished and save the lesson.
  - Click on the "Sentences - options" button, and then, for each sentence, select the radio button "Specific" (for Type) and use the "Set action file" button to find your video files in the "anim\ActiveVideo" folder – see below.



Sentences - options	Action File	Pre 1	Pre 2	Post 1	Post 2
The princess{s} kisses{v} the prince{o}	anim/ActiveVideo/PrincessKissesPrince.mpg	0	0	0	0
The princess{s} kisses{v} the princess{o}	anim/ActiveVideo/PrincessKissesPrincess.mpg	0	0	0	0
The princess{s} pushes{v} the prince{o}	anim/ActiveVideo/PrincessPushesPrince.mpg	0	0	0	0
The princess{s} pushes{v} the princess{o}	anim/ActiveVideo/PrincessPushesPrincess.mpg	0	0	0	0
The prince{s} kisses{v} the prince{o}	anim/ActiveVideo/PrinceKissesPrince.mpg	0	0	0	0
The prince{s} kisses{v} the princess{o}	anim/ActiveVideo/PrinceKissesPrincess.mpg	0	0	0	0
The prince{s} pushes{v} the prince{o}	anim/ActiveVideo/PrincePushesPrince.mpg	0	0	0	0
The prince{s} pushes{v} the princess{o}	anim/ActiveVideo/PrincePushesPrincess.mpg	0	0	0	0

- Close the Sentences – options dialogue, save the lesson, test run from within the Lesson Editor (make corrections, if necessary) and then in lesson mode in the main program.

Recording new words/phrases

You can record your ownsound via a microphone connected to the mic input of the computer. You can use the standard ”Sound Recorder” utility in Windows. It is generally found in the Utilities – Entertainment folder in the Start menu. If not, you could find it under the name of **sndrec32.exe** in the ”system32” folder within the WINNT or Windows directory. This utility will only supply you with very basic editing facilities, like trimming the start and end of the sound recording.

For better recording and sound editing features we can recommend the use of the free **Audacity** program (see http://audacity.sourceforge.net), or the sound editor utility of the Nero CD/DVD package, or any of several other sound editing products in the market (such as GoldWave, SoundForge etc.).

NOTE! The quality of sound recorded directly into computers is very varying. It’s a good idea to team up with somebody with basic knowledge of sound technology to achieve higher quality. It’s often better to record sounds on an external recording device and then transfer it to the computer - re-recording via the Line-in connector, or by direct transfer of ready sound files in a suitable way.

Sound files should be stored in the **wav** or **mp3** format, and should be placed in the ”media/audio/words-en” (etc) folder for recorded words/phrases for the lessons, alternatively in the ”media/audio” folder for effect sounds for the animations.

In general we recommend the technical quality of 44 100 kHz; 16 bits; mono or stereo, for the digital sound files.

Making new lessons available

The lesson menus of Omega-is are dynamically displaying the content of the ”active” folder of the ”lesson-xx” folder, where ”xx” represents the language; ”en” for English, ”sv” for Swedish, ”nb” for Norwegian (bokmål), etc.

To make a new lesson available in the lesson menu; browse to the sub folder of the “active” folder where the new lesson fits, and create a folder with a suitable name. In this you place the xxx.omega_lesson file. To represent the lesson by an image in its button, one or two image files called “image.png” and “image_enter.png” should also be placed in the lesson folder. (The image with the name ending with ”_enter” is the one displayed when the button is in focus).

The structure for all this is depicted above in chapter D, ” The ”lesson-en” folder - English lessons” section, page 27. Graphics used for “image” and “image_enter” files can be found in the ”LessonUI-Graphics” folder, within the ”media” folder.

Advanced – variables, stories etc.

The Lesson Editor supports several additional sophisticated functions for dynamic referencing from the student’s choices in the lesson to the corresponding animation files and actors within these. Moreover, there are functions for the transfer of information between a series of lessons – and choices made within these – which makes it possible to construct stories.

These functions can be somewhat complicated to explain and absorb. A rather short presentation will follow here. In addition, interested lesson authors are advised to study, learn from and re-use content and functionality in existing lessons and stories.

Variables - references

In a lesson, up to three variables (1, 2 and 3) can be set in the property dialogue for each word in the item buttons. These variables can be referenced from the fields for ’Text’, ’Sound’ and/or ’Action file’ in the properties for words in other item columns – or from the property dialogue for the target slots in the Sentence Line – of that lesson. Let’s call this ”internal” variable references. (Variables can also be referenced between the different parts of stories – ”external” variable references).

The format for internal variable references is {+1} for variable 1 in the next item column (to the right), {++3} for var. 3 in the second next, {-2} for var. 2 in the previous column (to the left) etc.

Many examples of the use of internal variable references can be found in the H3-H5 and J lessons.
For more details on external variable references in stories – see the “Stories” section below.

Linked animations

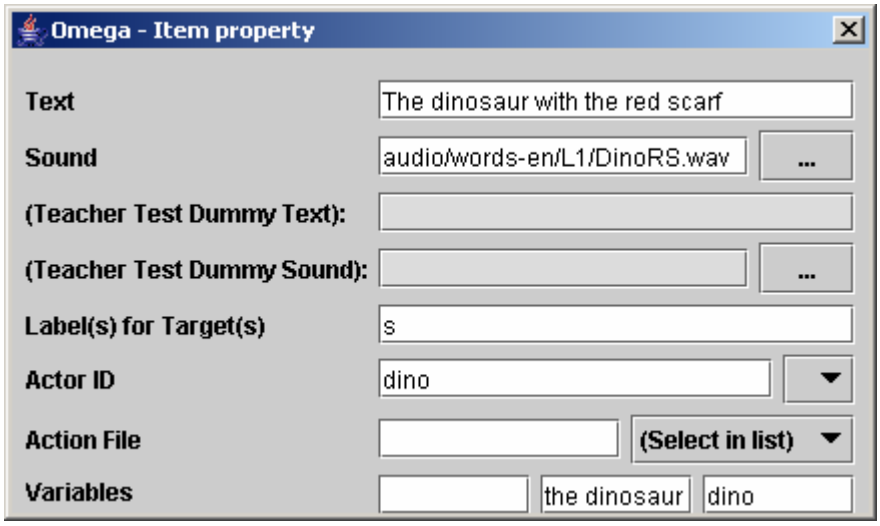
Normally each sentence in a lesson links to one single animation file. But it is actually possible to link to more than one animation, i.e. to play two (or more) animated scenes in a row, as an illustration of a sentence.
This can be accomplished in a couple of ways: One is to let two action words link to one animation each, as in the example ”Demo-LinkedAnim” lesson, found in the Lesson menu in the X-Demo–Xercises, in the ”lesson-en\active\X-Demo\Xercises” folder.
Another way is to reference several comma separated animation files in the action word properties (see the L3-8 story lesson in the L3 folder, under ”lesson-en\ActiveStories”). NOTE! The corresponding may also be needed for Actor and Path IDs.

Stories

When a story is constructed, there is a need of functions to retain the memory of previous choices of actors, environment etc., so that the story line can be maintained. In Omega-is this is achieved by the use of ”external” variables. In this way the information is transferred between the parts of the story. We’ll look at a few examples found in the accompanying L1 and L3 stories:

The first part of the L1 story – lesson ”L1-1” – is found in the ”lesson-en\active\L-Stories\L1” folder.

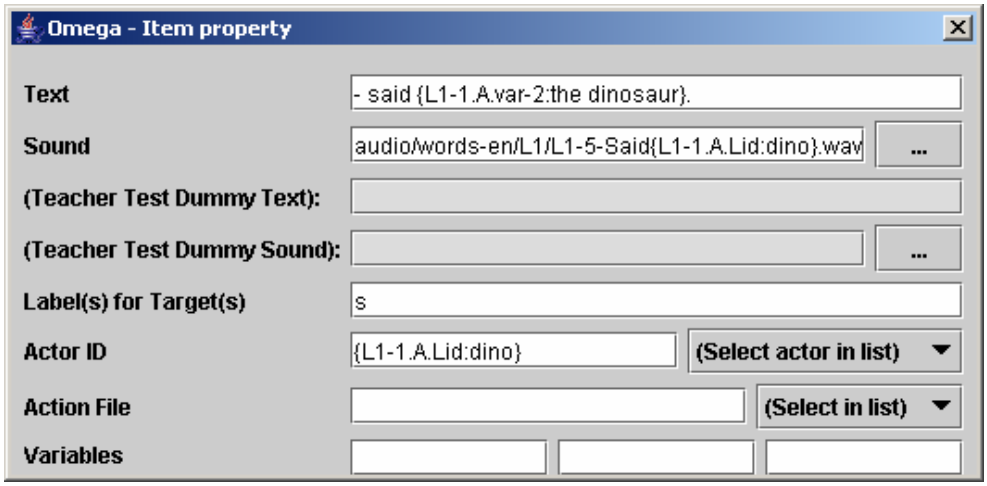
Let’s look at the following item properties for one of the two alternative main actors of the story – ”the dinosaur with the red scarf” (the other one is ”the dog with the yellow hat”):



Note the Actor ID (”dino”) and the two Variables ”the dinosaur” (var-2) and ”dino” (var-3)!

This lesson has got its ”Lesson name” set to ”L1-1”, and has been marked as ”First in story”.

With this in mind, we find our way to the ”L1-5” part further on in the story sequence. It is found in the ”C:\omega\lesson-sv\ActiveStories\L1” folder. There we take a look at the item properties for the phrase ”- said the dinosaur”:



The content in ”Text” is ”- said {L1-1.A.var-2:the dinosaur}.”

This means that variable 2 (”var-2”) for the actor at Path ID ”A” in ”L1-1” is called. If this variable content is not available, the value after the colon sign will be used i.e. ”the dinosaur”. The parts of the variable reference are separated by full stops/periods, and are all contained between curly brackets.

If ”the dog with the yellow hat” is selected in L1-1, the value of ”L1-1.A.var-2” will be ”the dog”. The text shown in the text button will then come out as (instead of the cryptic variable reference) : ”**- said the dog**”.

Also take a look at the variable reference in ”Actor ID”: There the Actor ID – called ”Lid” – of the actor at Path ID ”A” in lesson ”L1-1 is called”. Default is ”dino” (after the colon), but it would in our example above have been ”dog” that is the Actor ID of the dog.

The ”Lid” variable is also used for ”Sound” to play the sound file that matches the text representation of the actor that has been chosen in the story.

It is obvious that it’s a bit of high level programming to set up all variables and references to make a story run as intended. It may become rather complex, especially in a more elaborate story than this one. The interested story designer may dive into the variable referencing of the L3 story to get a closer look.

There we can find references like:
”anim/Stories/L3/L3-5{L3-4.W4.var-3:meadow}” – which means a call to the animation file named “L3-5…” – with the addition of variable 3 (var-3) of the selected item in word column 4 (W4) in lesson L3-4, and that the default addition is ”meadow” (i.e. ”L3-5meadow”).

Functions and commands

Here is a list of the commands and functions of the Omega-is Lesson Editor:

File:

New – opens a new lesson template file

Open – displays the Open file dialogue – for the opening of an existing lesson

Save – saves the current lesson file under the current name

Save as – displays the Save file dialogue – allowing the entering of a files name

Exit – exits the Lesson Editor – via a control question.

Help:

- **Manual** – displays chapters D, E and F of this manual in the default browser
- **About Omega** – shows information about Omega-is in the default browser
- **About Lesson Editor** – as above

Buttons / functions of the Tool bar:

Story functions:

- **Lesson name** – option to name the lesson – used for referencing between part lessons of stories
- **Next in story:** – used for linking to the next lesson file of a story sequence. The check box is ticked, and the next lesson file is found via the Browse button "…"
- **First in story:** – the check box is ticked if the current lesson file is the first in a story sequence. Such a first story lesson file must be placed in its own folder under L_Stories in the "active" folder – under "lesson-xx".
  The other files of a story should be collected in a common folder in the "ActiveStories" folder.

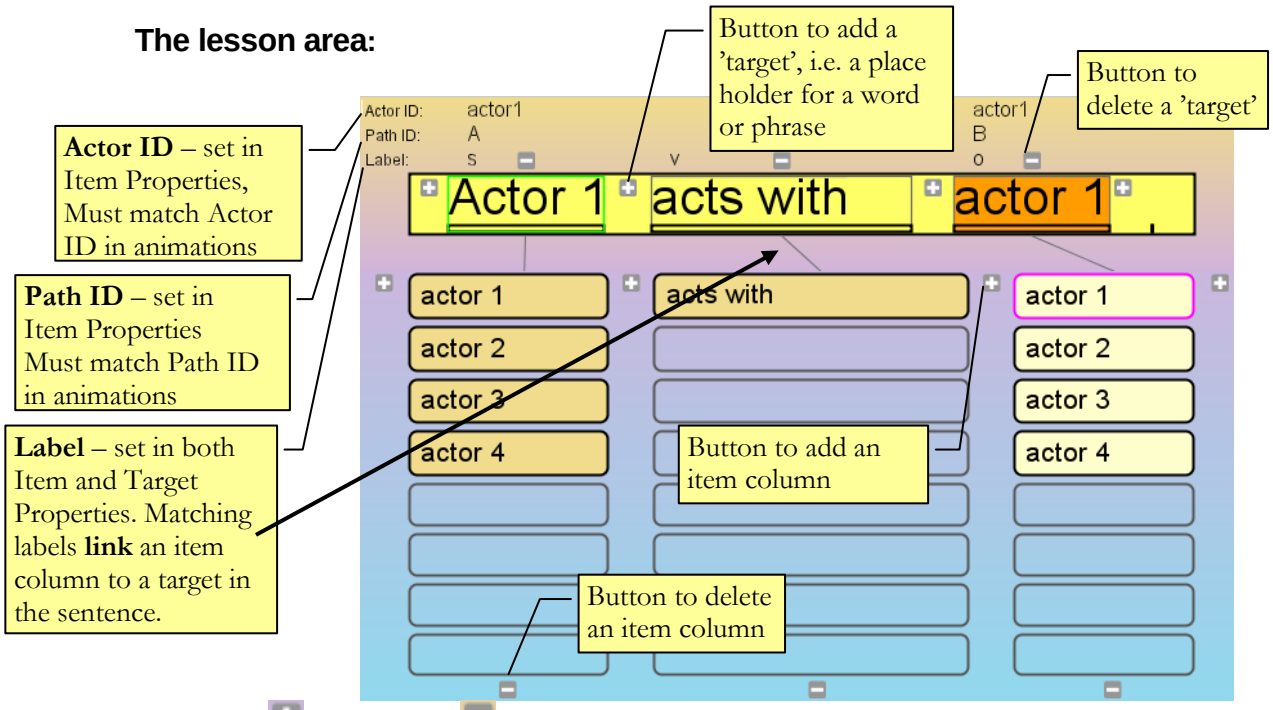
Properties buttons:

- **Target – in sentence** – opens the properties dialogue for the target slots in the Sentence Line – can also be opened by **right clicking** a box in the Sentence Line.
- **Item – in list** – opens the properties dialogue for the words/phrases in the item columns – can also be opened by **right clicking** an item button in the columns.
- **Sentences – options** – opens the window for sentence properties, i.e. the list of the possible sentences/combinations that can be created from the words/phrases in the lesson – with the option to link sentences to video clips, and to select specific sentences for specially prepared pre- and post-tests in Teacher Mode.

Command buttons:

- **Redraw** – redraws the window – may sometimes be needed after the addition or deletion of Item columns or Target boxes – for a proper display etc.
- **Play** – plays/runs the animation that represents the current sentence created in the Sentence Line. (Requires that a complete sentence has been created of course.)
- **Listen to sentence** – plays the sound files representing the words/phrases in the current sentence in the Sentence Line.
- **Edit animation** – opens the Animation Editor with the animation linked by the 'action word' of the current sentence (possibly influenced by references in other words).

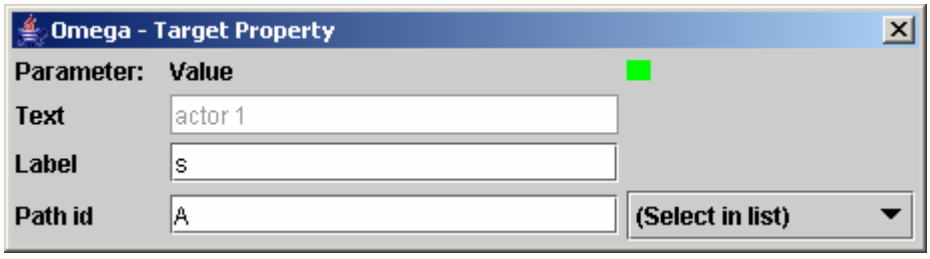
The lesson area:



Plus- and **Minus-** buttons by the Sentence Line and item columns (see above) add or delete 'target' or 'item column', respectively – after a control question. **NOTE** - maximum 6 word/phrase targets in the Sentence Line!

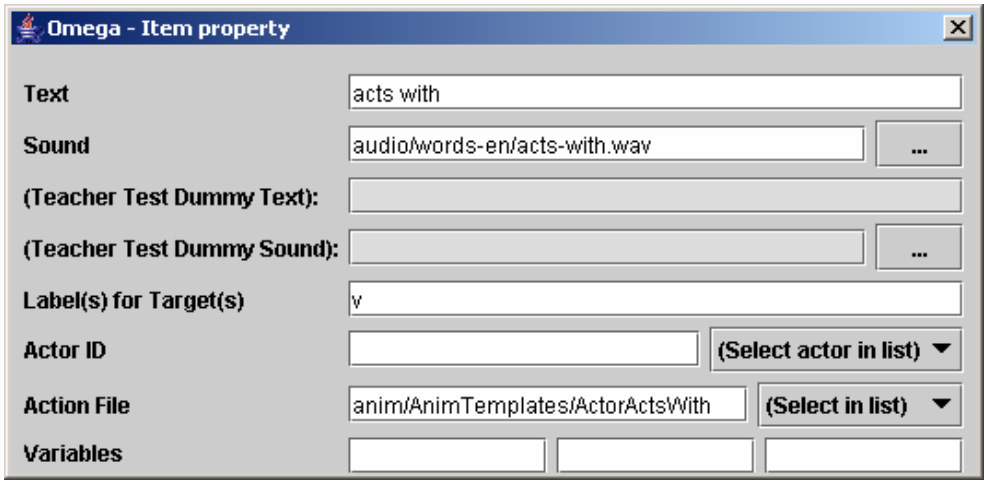
Property dialogues:

Target – in sentence:



- **Text** – displays current selection for the target (if any) – no text entry!
- **Label for Target** – should be set to match the item column that should be linked.
- **Path ID** – Selected from the available for desired role in current animation(s). **NOTE!** May refer to a variable in another item column for choice of Path ID (e.g. lesson H4).

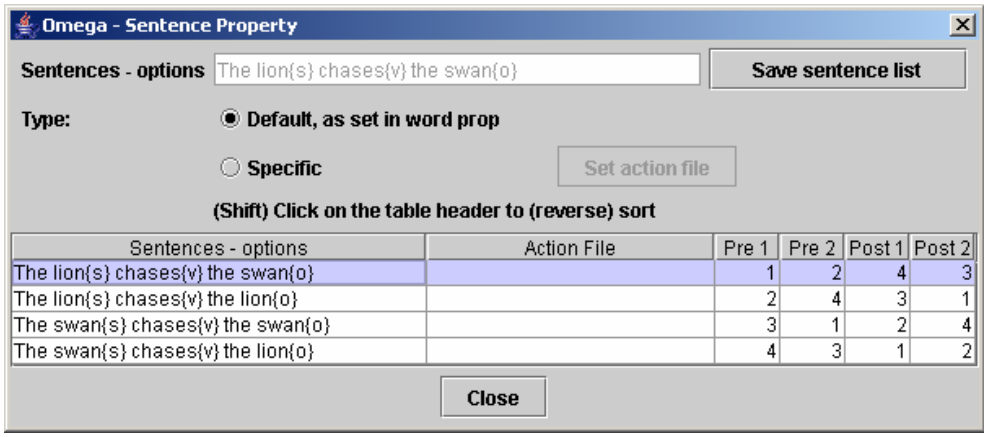
Item – in column:



- **Text** – the word or phrase to be displayed and selected for inclusion in the sentence.
  NOTE! The item may contain references to variables in other item columns that modifies what is displayed and sent to the Sentence Line.
- **Sound** – the sound file that matches the text.
  NOTE! The file reference may contain variables from other item columns that modifies the reference – see below under ”Variables”.
- **(Teacher Test Dummy Text):** – optional dummy word (distraction word) for post tests in teacher mode.
- **(Teacher Test Dummy Sound):** – sound file matching the dummy word.
- **(Label(s) for Target(s)** – here a label is entered to match the ”target” in the sentence that should receive the selected word/phrase.
  NOTE! The label represents the whole item column - needs only to be entered once.
  NOTE! Two labels may be entered – separated by commas – to link an item column to two different targets in the Sentence Line – as in lessons F3, F9 etc. where the nouns can be both the subject and the object in the sentence.
  NOTE! Several item columns may have the same label, so that they enter words to the same target in the Sentence Line – as in the 1-word lessons B3 and B4.
- **Actor ID –** selected to match ”actor” items to the corresponding aktor in cast of the animation that has been linked via the “action” word (including optional variables from other words). Valid actors are found under ”(Select actor in list)” – provided a sentence has been created in the Sentence Line, so that an animation is referred.
  NOTE! Actor IDs are not needed for words that should not refer to any actor in the animation, such as ”action” words (generally verbs) and words with other functions.
- **Action File** – for action words (typically verbs), here the animation file should be referenced that represents the implied event of the action word (plus optional variables).
  NOTE! The file reference may contain variables from other item columns that modifies the actual reference – see below under ”Variables”.
- **Variables** – Up to three variables (1, 2 and 3) can be entered for each item. These may then be referenced from the fields for Text, Sound and/or Action File in the properties for items in other columns in the lesson (”internal” variable references), or from other lessons/parts in stories (”external” variable references).
  The format for referencing variables internally is; {+1} for variable 1 from the next item column (to the right), {++3} for var. 3 in the 2:nd next, {-2} for var. 2 in the previous column (to the left) etc.

For more details, and about external variable references in stories – see the previous section above about "Advanced – variables, stories etc.".

Sentences – options:



This dialogue displays a list of all the sentences that can be created from the available words or phrases in the lesson – for overview and documentation, for optional linking to video clips, and for special selections for tests in Teacher Mode.

Save sentence list – saves the list to a text file, for documentation and printing from a text editor application.

Default, as set in word prop – the default setting, meaning that the sentence will be represented by an Omega-is animation, as set in the properties of the lesson material.

Specific – Set action file – means that the selected sentence will be directly linked to, and represented by, a short mpg or mov video film.

The numbers in the list under Pre 1 and 2, Post 1 and 2 – control which sentences will be presented in Pre- and Post-tests in Teacher Mode, and in which order.

Chapter F

Chapter F:
The Animation Editor

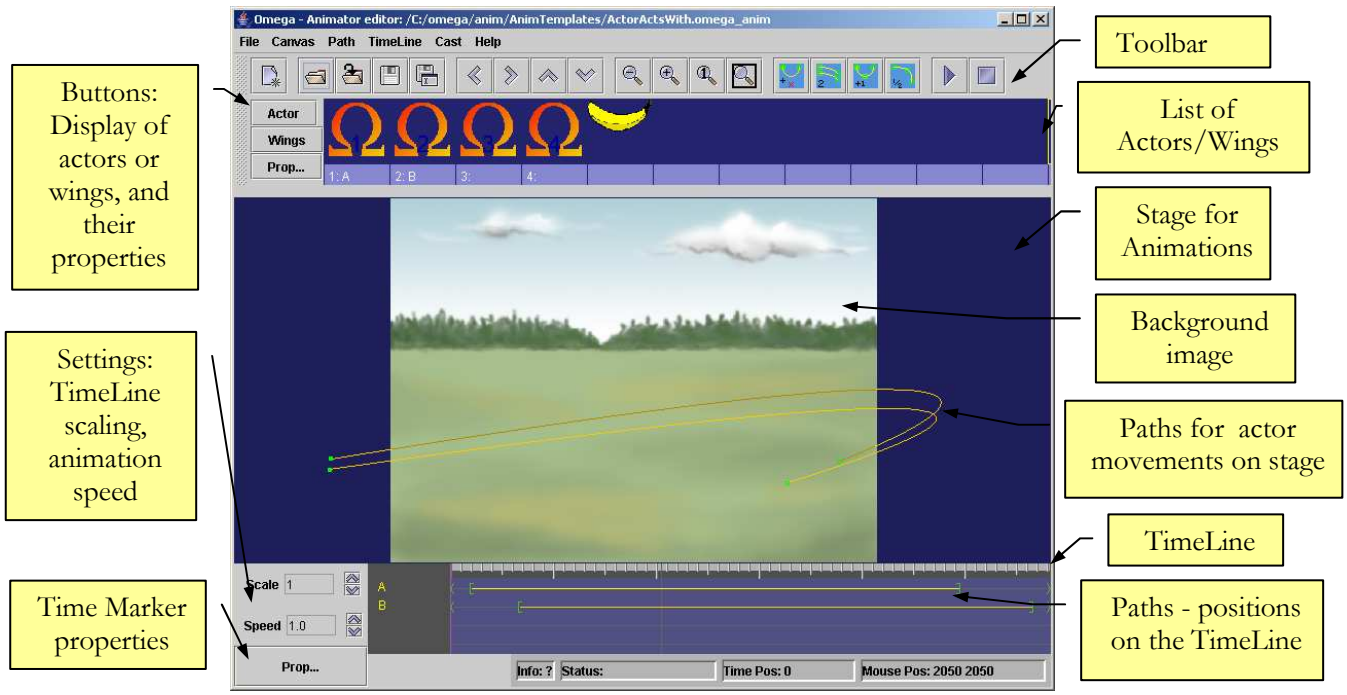
Overview

The Animation Editor is a tool allowing the creation of new Omega-is animations, which can then be linked to the language material in the Lesson Editor.

The Animation Editor also makes it possible to edit existing lessons. An alternative image may be used, or the speed of some of the actors in the animation may be adjusted. You may want to use an alternative background in a specific lesson. Perhaps the use of a photo of the student's school yard might be more motivating.

NOTE! The Animation Editor keeps track of changes made in an animation! If they haven'tbeen saved, this is stated in the "title bar" of the window, and a warning is given when the commands "Exit", "Open" or "New" are selected..

First a look at the Animation Editor window and its parts:



The Animation Editor is an application on its own that may be opened, either by selecting ”Edit animation” from within the Lesson Editor (see the previous two chapters), or by selecting the **Animation Editor** from the Start menu.

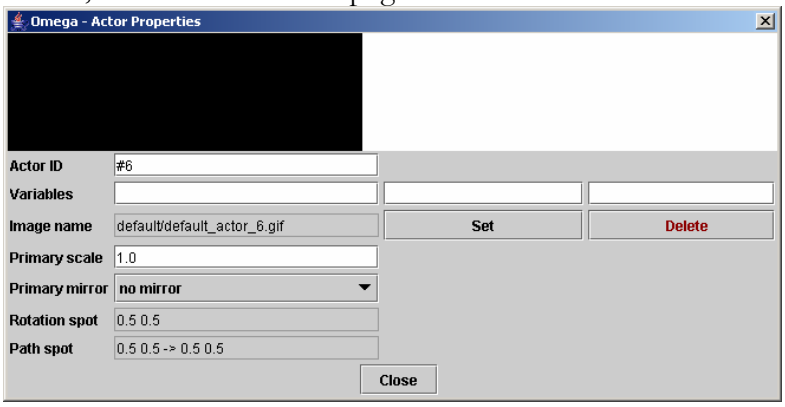
Minor changes in existing animations

Minor changes in, or additions to, existing animations is a good way to get going with using the Animation Editor.

Adding new actors to an animation

- Start the Lesson Editor.
  - Open the lesson file ”HeadlessHunt.omega_lesson” that is used in the ”Xercise-Demo2” lesson in the” Xercises” folder in the ”X-Demo” lessons. We will add two new swappable heads to this wild hunt animation.
  - Click on some words to create a complete sentence in the Sentence Line.
  - Then click on the ”Edit Animation” button – the Animation Editor will open with the” HeadHunters.omega_anim” animation (– in the /anim/MyAnim folder).
- Maximise the window. In the actors list you will find two headless actors, plus four heads (belonging to the initiators of this program) – see below. We will now add two more disputable figures to the list. Right click on the first empty slot in the Actors list (or left click on the slot, and then on the ”Prop...” button).

- An empty Actor Properties dialogue is displayed – click on the ”Set” button to link up to a new actor image. Open the ”actor”, and then ”MyActors” folders, and select the ”bush.png” file.

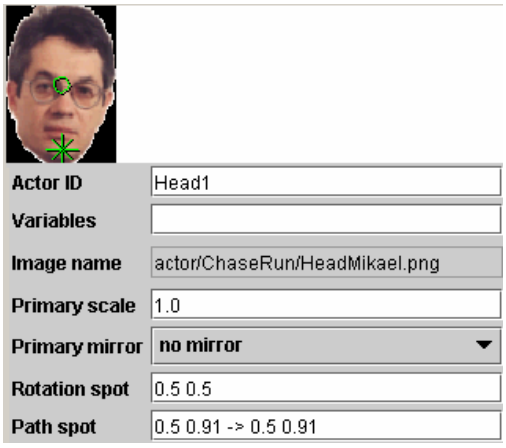


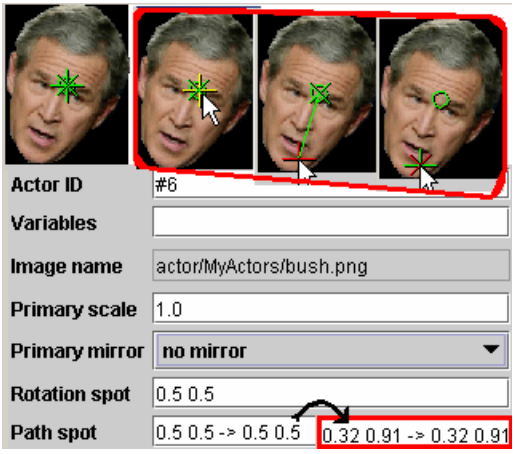
- Repeat the procedure for the next actor slot, and select the ”saddam.png” file. The actors list (cast) should now look as follows:

Grab the label ”2:A” (with the mouse) and drag it to the Saddam head, as shown above. Likewise drag the ”4:B” label to the Bush head.

- Close the Actor Properties dialogue to get a free view. Now test run the animation with the two new actors: Click on the Play button! The result shows we’re getting there, but that some adjustments are needed. Bush’s and Saddam’s heads are slightly displaced in the animation, and possibly big (but this may be regarded as a good point …).

- Re-open the Actor Properties dialogue by right clicking on one of the heads, e.g. the first one. The properties for the actor with the Actor ID ”Head1” are displayed (below left). Then click on the Bush head/actor and compare:





As shown there are three green markers in each image – a circle, a cross and an X. The circle represents a ”Rotation point”, currently of minor interest. The cross and the X represent so called ”Path Points”, i.e. points at which the actor is attached to its path in the animation – at the start and the end. In ”Head1” these path points are both placed at the actor’s chin. In the Bush head all three markers are in the middle of the forehead. Now move the ”path points” (cross and X) according to the extra images in the red frame above. When you point the mouse at the markers, one is highlighted in yellow. If it’s not the desired marker, then move the mouse cursor away and back again – then the next marker is highlighted. Use the mouse to grab and drag the highlighted cross marker to the chin area, as shown in the image above. Repeat this for the X marker. The numeric values for the path points will change accordingly, to end up at the approximate values to the ones in the image above. Repeat the whole procedure for the Saddam actor!

- Test run the animation again. Satisfied? You could possibly try changing the ”Primary scale” (size) of the new heads, e.g.. to the value of 0.8. NOTE that

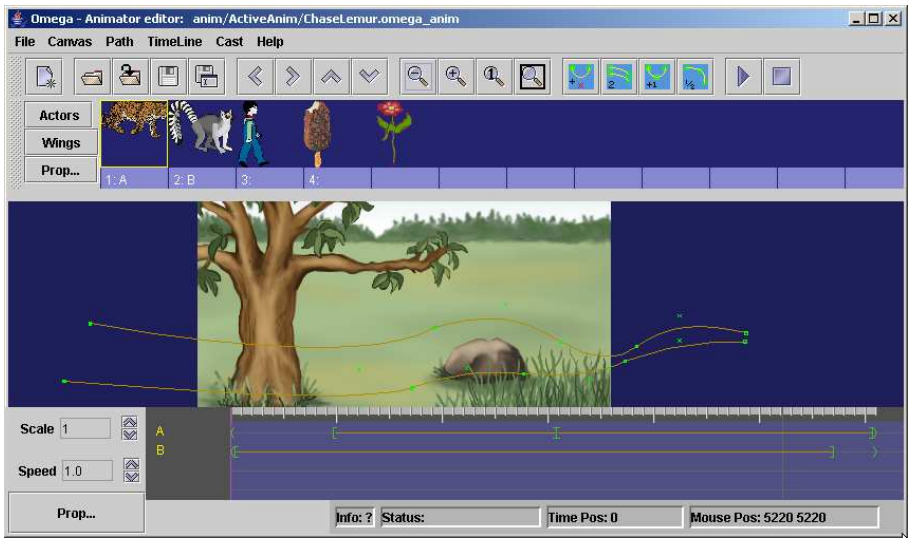
decimal points – not decimal comma – must be used for all parameters in the Animation Editor!

- Finish the editing of this animation by replacing the anonymous #6 and #7 Actor ID values by suitable references of your choice – e.g. ”Bush” and ”Saddam”, or ”Head5” and ”Head6”. Save the animation, and close the Animation Editor.
- You will now be back in the Lesson Editor, and you may add the words ”Bush” and ”Saddam” – with references to the new actors in the animation. (See the previous chapter for details about how to do this – matching sound files will be found in the ”extra” folder.).

Changing the length of animations

A certain animation may be perceived as being too short/fast – or too long/slow – for a certain student. This may motivate an adjustment of the length of particular animations. This is how to do it:

- Start the Lesson Editor, open the lesson file ”demo.omega_lesson” in the ”X-Demo” lessons in the ” Xercises/Xercise-Demo1” directory. Create a sentence containing the word ”chases”. Then select ”Edit animation”. The Animation Editor with the ChaseChased.omega_anim will appear, approximately as below:



- At the bottom of the window is the ”TimeLine” where the animation paths are represented. In this animation these are covering a time span of about 6 seconds. We will now compress the action time to about 4 seconds. First test run the animation (in a maximised window) to get a feeling for the original speed/length. Now test to shorten the length on the TimeLine of the up most path; Grab, by **Ctrl + mouse click and hold**, the end marker of the path (”]”), and drag it to approximately 4,1 seconds position, as shown below.

- Then repeat the procedure for the lower path (to around 3,9 sec), and finally shorten the whole animation time by **Ctrl-dragging** one of (and thereby all) the end markers ”>” to an approximate 4,2 sec position, as below.
  (NOTE! A minor bug makes the end marker jump to the right around 0,8 sec by the Ctrl-mouse click. Disregard this, and the resulting deviation between the mouse pointer and the end marker, and drag the end markers to the desired position as in the figure below.)

- Test run the animation again. Was the result to the better or worse? Save the animation if the result of the change is regarded positive. Otherwise, feel free to continue to play around with the length of the paths etc, and test the results. Then close the Animation Editor – after saving, if desired.

Changing the background image

We will continue with the same Xercise-Demo1 lesson as in the previous exercise.

- Now create a sentence containing the phrase ”jumps over”. Click on the ”Edit animation” button. The Animation Editor will appear, as below, with the ”JumpOver.omega.anim” animation (in the /anim/MyAnim folder):

- We will now change the background of this animation. Open the ” Canvas” menu and select ”Set background”. Open the ”background” folder and select the ”schoolyard1.jpg” image. This could of course just as well, or even better, be a photo of the local school yard or classroom.

- The scene will now look something like the image in the next section below. We will there do some necessary adjustments of the positions and form of the animation paths.

Changing the positioning of the animation path – in the scene

The image below shows the result of the change of background image in the previous exercise. A test run of the animation (in a maximised window) will show that the animation paths are now positioned a bit too far up in relation to the new background. We'll now move the two paths slightly downwards on the stage.

- Move the mouse pointer, as shown in the above image, towards the encircled group of green points in the jumped-over actor's contracted path. As shown, the form of the mouse pointer changes when it comes close to the path points – from a hand to a positioning cursor.
- Press and hold the **Shift** key, and (**click-**)**drag** this whole path, and its path points, downwards with the mouse – as the blue arrow indicates.
- Make a corresponding repositioning of the stretched and multi section path of the jumping actor – by **Shift-dragging** one of the points of the path downwards in relation to the background – about as far as in the previous operation.
- Now test run the animation to see the result. (Unfortunately it may be somewhat difficult to see the bottom part of the animation as the scene is partly obscured by the tools of the Animation Editor. You will have the full overview of the results when the animation has been saved and run from within a lesson – see below.) Do further adjustments of the positions of the paths, as needed, save the animation and close the Animation Editor.

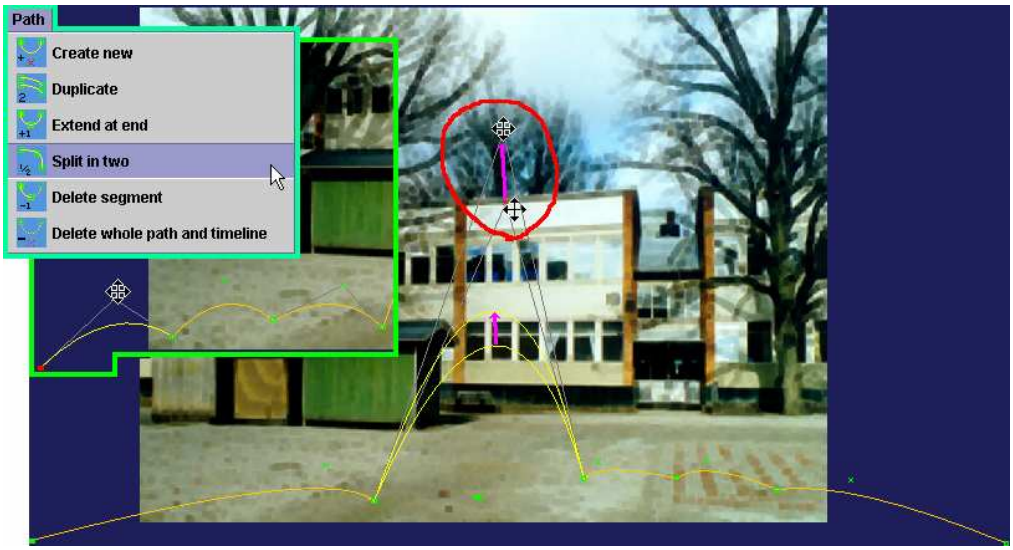
Back again in the Lesson Editor:

- Now the "Xercise-Demo1" lesson can be test run with the two newly modified animations. If something is unsatisfactory, just use the "Edit animation" button to return to one or both of the two linked animations for further editing in the Animation Editor – as in the following section below …

Changing the shape of animation paths – in the scene

We'll stick to the Xercise-Demo1 lesson, and return to the "jumps over" animation for some additional alterations of the path of the jumping actors.

- Create a sentence with "jumps over", and select "Edit animation". We start by increasing the height of the jump over the static actor:
- The scene of the animation should, After our previous alterations, look something like the image below. Move the mouse pointer up over the path to find the small green x-shaped point representing the "curve handle" that determines the curve of the jump. The mouse cursor changes from a hand to a "positioner" when it approaches the handle (the lower position in the red encircled area of the image.) When you click on the x-shaped handle point it is selected and coloured red.
- Now grab the handle (with the mouse) and drag it upwards – as the purple arrow indicates. The bow of the jump will adjust accordingly, as the image shows. Test run and adjust further until the result is satisfactory.



We'll now make the appearance of the jumping actor a little "jumpier" by adding a few extra bows to the path:

- Start by clicking-selecting the starting point of the path – the filled small square bottom left dot – so it becomes red coloured (as in the inserted part of the image above).
- Open the "Path" menu and select "Split in two" (cyan framed insertion in the above image). The first segment of the path will now be split into two segments. Repeat the procedure for the first new segment. We will now have three segments before the jump.
- Now bend the three segments by dragging their x-shaped curve handles – as indicated in the above image. Test run, adjust if needed, and save.

There are of course many other things to discover, investigate and tweak in this "jumping-over-animation". For example, take a look at the properties for the different markers, or "trigger points", along the two paths on the TimeLine.

- Open the marker properties dialogue by clicking on the "Prop…" button at the bottom left of the TimeLine (or right click on one of the paths on the TimeLine and select "Marker properties").
- Then click on (select) the different markers, and study their properties, to see what's going on in each point. We will not now further study the different functions and how they can be utilised. We will return to this in the following sections. But a quick look at the features and properties of this animation gives some insight in some of the possibilities – to play sound effects, to scale (change the size of) actors, to flip "layers" (in front - in between – behind ...), to rotate actors etc.

Creating simple new animations - basics

We will now go through the creation process for a brand new Omega-is animation: We choose to let the stage present a space scene – with a moon and a comet. All necessary media material is included in the installation:

Starting the Animation Editor

- Quit Omega-is or the Lesson Editor, if any of them is running
- Start the Animation Editor from the "Animation Editor" start icon
- Maximise the window to full screen
- It is a good idea to start by saving the anonymous new animation under a suitable name. Select File – Save (or Save as – no difference at this stage). Save the animation in the "anim\MyAnim" folder e.g. under the name "MyMoonComet".

Selecting the background

- In the "Canvas" menu, select – "Select background". The Open file dialogue will open, displaying the content of the "media" folder.
- Open the "background" folder. Find and open the "comet.jpg" file.

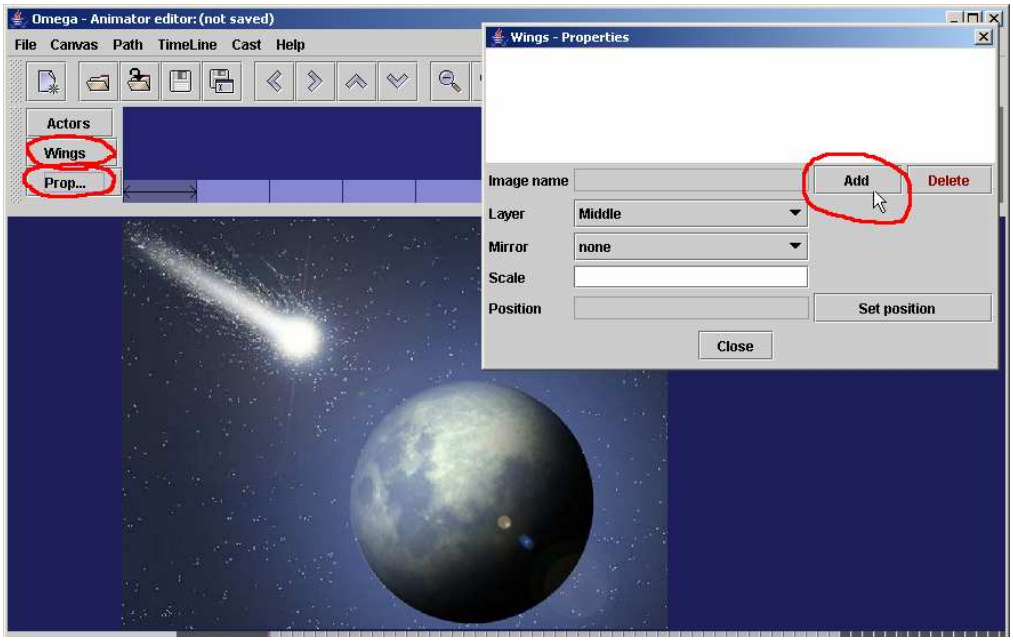
Save the animation again after this step, and then again after each further major addition or change!

We have now set the stage for our space animation. We will now soon let some actors enter and perform on it. But first:

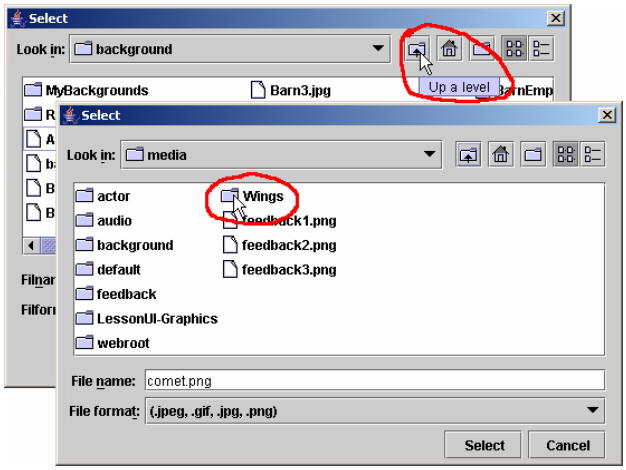
Adding and adjusting 'wings'/'props'

To provide an illusion of depth in the scene, it would be fine to be able to make the actors move both behind and in front of the moon and the comet. We achieve this by placing two wings in front of the moon and comet in the background image – copies of these two objects.

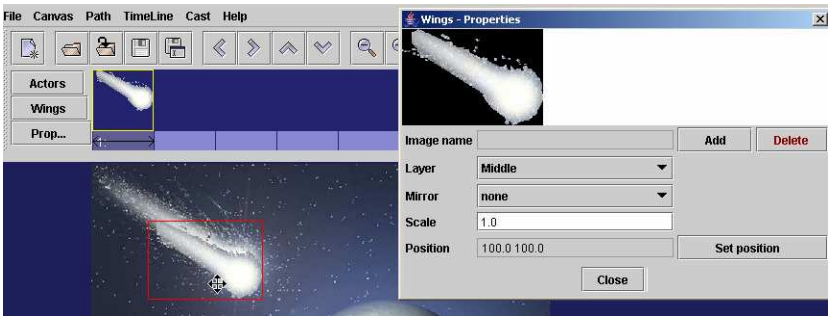
- Click on the "Wings" button, to the left of the Actors/Wings list.
- Right click on the first empty slot in the wings list – or click on the "Prop…" button – so that the "Wings – Properties" dialogue is opened – see the image below.



- Now click on the "Add" button to select a wing image file. We will find ourselves in the Open file dialogue again, which displays the content of the "background" folder as that's where we were last looking. Step back / up in the directory tree to the "media" folder, and then into the "Wings" folder. Select the image file "comet.png" (see the image to the right)



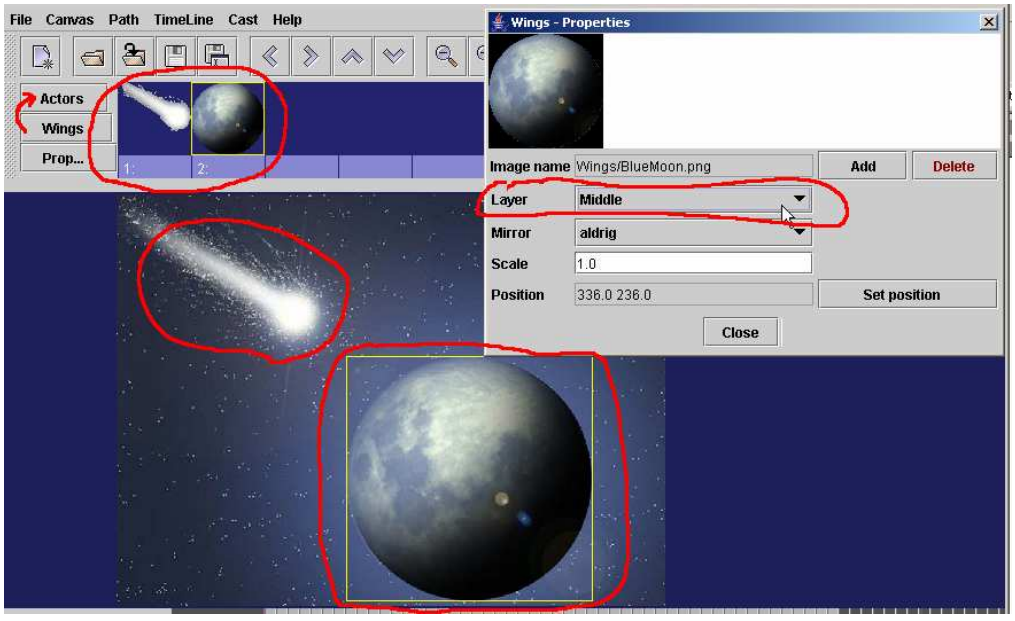
As shown in the image below, a copy of a part of the comet appears within a red frame. While the frame is red you can grab the wing with the mouse and drag it to a desired position on the stage/background – in this case this is on top of the comet in the background, so that it invisibly blends into the background image. NOTE; you can only move the wing once at a time. As soon as you drop it in a new position the frame goes yellow and the wing is immobile. To make it moveable again you have to click on the ”Set position” button in the ”Wings – Properties” dialogue.



- Then click on the list button for ”Layer” and select ”Middle” – see images above and below.

- (In ”Wings – Properties” dialogue, as you can see, you also have the possibility to change the scale/size of the wing/prop, and also to mirror it horizontally or vertically. But this is of no interest to us in this case.)

- Click on the next empty slot for a wing/prop (to the right of the comet). The ”Wings – Properties” dialogue will show up empty. Fill it with new content by repeating the procedure described above – click on ”Add” (we’re now already in the right folder) and select the wing/prop ”BlueMoon.png” – place it on top of its original – change Layer from ”Top” to ”Middle”. The screen should look something like below:



- Finish by closing the ”Wings – Properties”, and return to the display of the actors list by clicking on the ”Actors” button (red arrow in the above image).

Adding and adjusting actor paths – on stage

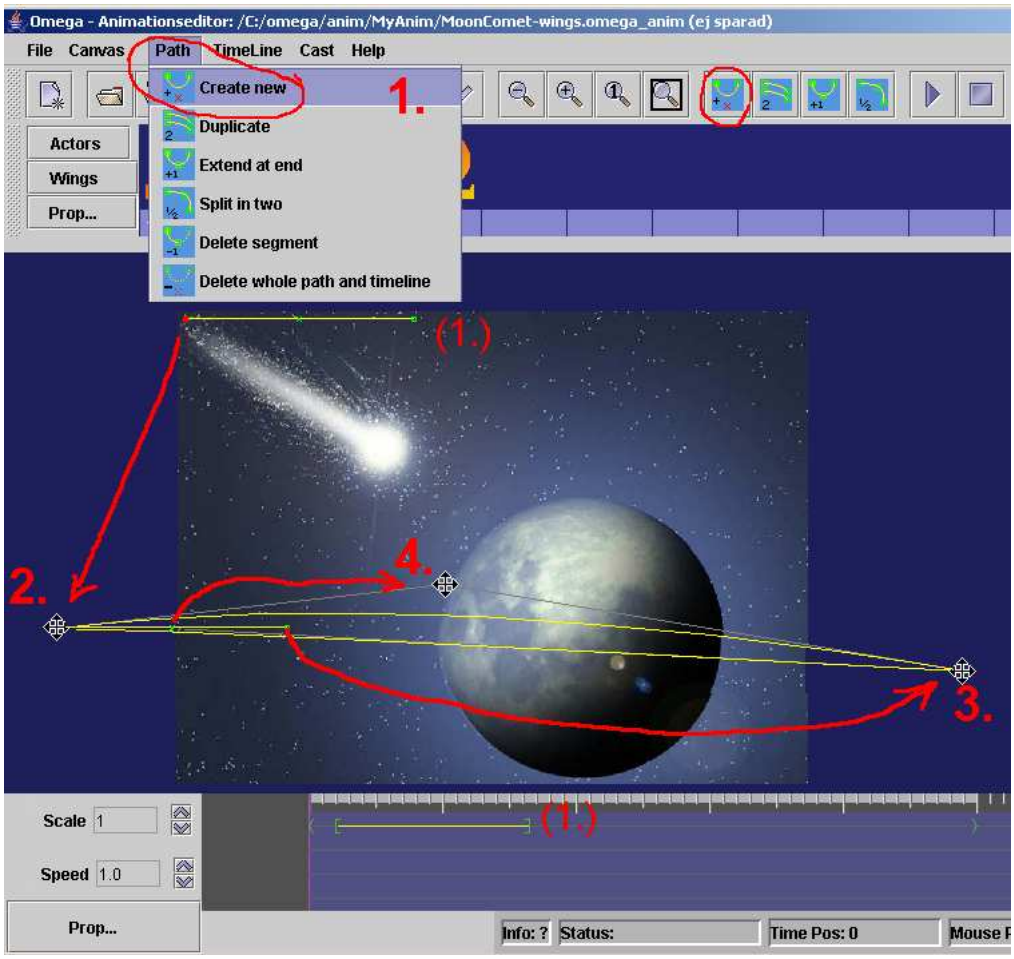
Now it's time to enter actor paths into the space scene:

1. Add another actor path by selecting "Path" – "Create new" from the menu, or by clicking on the corresponding symbol in the toolbar (see image below). The new path appears in the upper part of the stage, and with another representation down on the TimeLine.

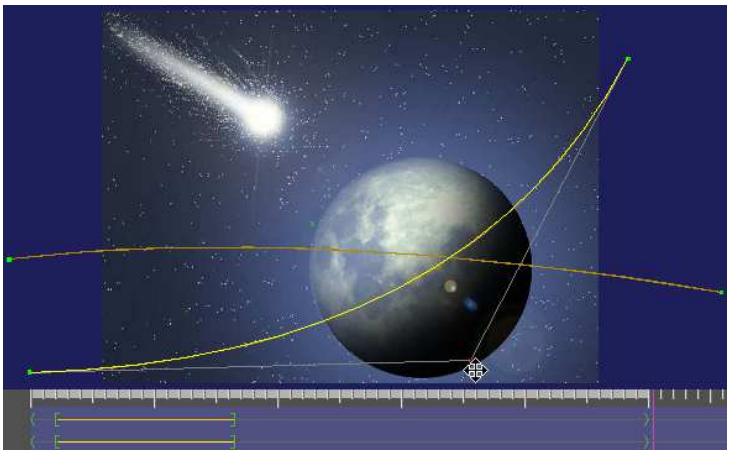
2. Press and hold the Shift key and use the mouse to grab the starting point of the path (on the stage – the left filled tiny square) and drag it – and the whole path – downwards to position 2. (approximately) in the below image.

3. Then use the mouse (NOTE; now without Shift!) to grab the end point of the path (the right un-filled tiny square) and drag it to somewhere around position 3. in the below image.

4. Finally, for this path, use the mouse to grab the "curve handle" (the small x) and adjust the curve of the path by dragging it to a position close to position 4. in the image below.



- Now add another actor path by repeating the process above, so that the result looks something like the image below.



- Test run the animation. The result may not be too impressive: Two of the Omega actors quickly sail by across the stage behind/beyond moon. We'll shortly try to improve the animation with new actors and other adjustments.

Adding and adjusting new actors

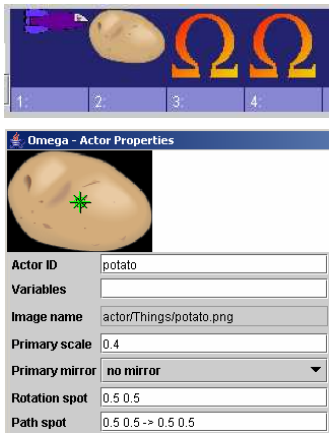
We continue with selecting some more suitable actors for our scene:

- Right click on the first slot in the actors list (or left click on the slot and then on the "Prop..." button). We'll replace the Omega actor with a rocket ship.

- Click on the "Set" button, and then browse to the "(media\)actor\Things" folder and in that select the "rocket(.png)" image.
  NOTE that there are additionally four images named "rocket-01", "rocket-02", "rocket-00", and "rocket-NoAnim! The number suffixes mean that the rocket actor will have an **"internal" animation**. This is achieved as the program will flip between the images with the -01, -02, and so on until -00 with a default speed of 0.2 seconds per image. More about this below.
  It's time to enter a suitable "Actor ID" for the actor – e.g. "rocket". Further possible adjustments of the Actor Properties can wait.

- Now click on slot number 2 in the actors list, and replace, as before, this Omega actor with the "potato.png" actor (from the same folder - "(media\)actor\Things").
  Give this actor too a suitable ID, and set its "Primary scale" to "0.4 ".

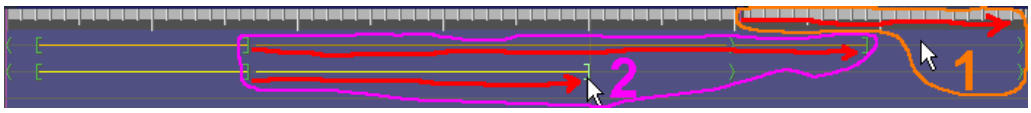
- Close the "Omega – Actor Properties" and test run.



Adjusting paths and events on the TimeLine

Now it's time to do some adjustments of how the animation is performed in terms of timing, and of how the actors relate to each other, and to the wings/props and the background:

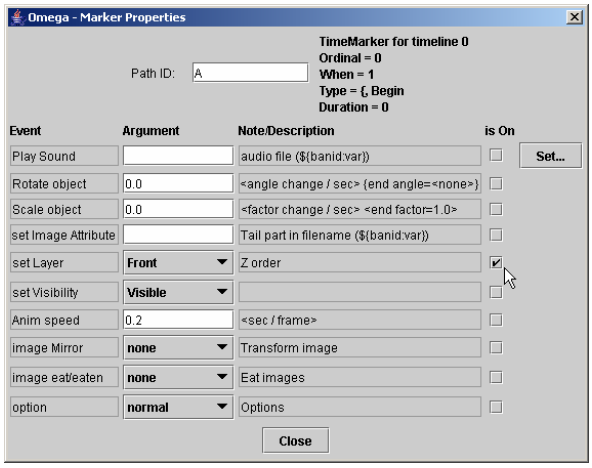
- The animation is very short and quick. We'll start by prolonging the time span.
  1. Press and hold the **Ctrl** key, and use the mouse to grab and **drag** the right end of the grey marking (and the right angle markers) – from the 5-seconds position to the 7-seconds position according to 1 in the image below.
  2. **Ctrl-drag** the right end markers of the two paths on the TimeLine, so that they are lengthened as shown at 2 in the image below.



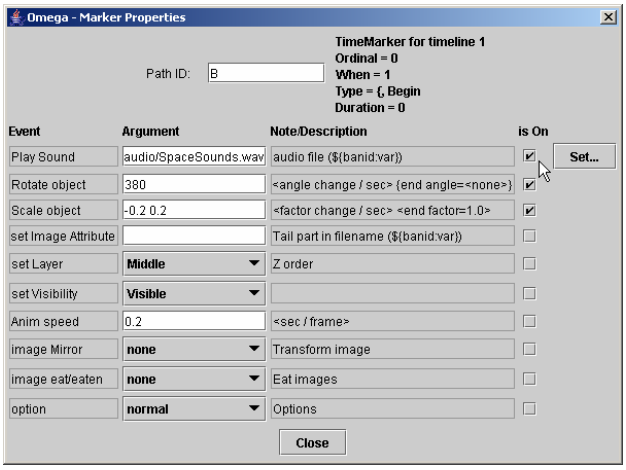
  3. Now press and hold the **Shift** key, and **drag** the whole upper/first path to the right – i.e. later – on the TimeLine, as in the image below. Test run!

Both the actors are now moving somewhat slower – the potato/asteroid slightly ahead of the space ship. Now we'll further fine tune the behaviour by adding events and properties for the actors on the TimeLine. (Save the animation!)

- First some basic adjustments of the Rocket's behaviour:
  1. Open the "Marker Properties" dialogue – click on the "Prop…" button to the left of the TimeLine (or right click on one of the markers/paths on the TimeLine and select "Marker properties" from the pop-up menu).
  2. Now click on the start marker (the little green left angle "<") for the first/upper path on the TimeLine. The properties for that specific marker are displayed.
  3. First give this path a "Path ID", by entering an "A" in the Path ID field at the top of the dialogue (see the image to the right).
  4. We'll now make the rocket fly in front of, on the near side of, the moon:
     Click on the list box at the "set Layer" label, and select "Front".
     Also click/check the little check box to the right of this property – as in the image to the right.



- We'll now do the corresponding adjustments for the other path – that of the potato:
  1. Click on the "Prop…" button to the left of the TimeLine, and then on the start marker of the second/bottom path, so that its properties are displayed.
  2. Set a "Path ID" also for this path, for example "B".
  3. We'll now make the actor of this path rotate. This is accomplished by entering the 'argument' "380" (degrees per sec.) for the rotation speed in the field next to "Rotate object" – and by activating the function by checking the "is On" checkbox.
  4. We'll also make our actor gradually become smaller by entering the argument "-0.2 0.2" in the field next to "Scale object". NOTE the Space between the two decimal numbers! This means that the object will become smaller (in relation to its primary scale/size) with a factor of –0.2 (a negative value will scale down, and a positive value will scale up), and down to a final scale of 0.2 of the primary scale/size.
  5. Lastly we'll start the playing of an effect sound by clicking on "Set…" button next to the "Play sound" property. We'll browse in the "media\audio" folder to find the sound file called "SpaceSounds.wav/mp3". Check the "is On" check box! The result should look like the image below. Test run and save!

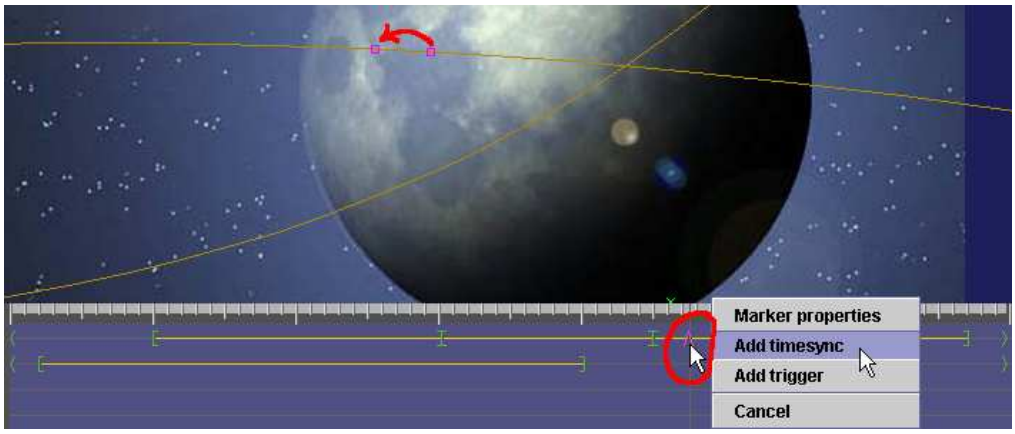


- We'll now add a few new **trigger** markers in path A – to see how "Image Attributes" are working:
  1. Right click on path A (on the TimeLine) somewhere around the 3-seconds position – select "Add trigger" in the pop-up menu – a new marker will appear in path A.. Make sure that the Marker properties dialogue is displayed, and click on the new marker to display its properties.
  2. Enter the text "NoAnim" in the field next to "set Image Attribute", and check the "is On" check box . This means that the "rocket.png" image (with its internal animation) will here be swopped to "rocket-NoAnim.png", resulting in the rocket engine looking like it's been shut off in the animation.
  3. In the same manner, add another trigger marker to path A – around the 4.5 seconds position – and check the "is On" check box next to the "set Image Attibute" property – NOTE; without entering any attribute text. In this way

the Image Attribute is re-set to zero, and the animation of the rocket/engine starts again.

- We'll round off by adding another type of event and marker to path A, a so called "**timesync**". This one will be represented by a marker on both the TimeLine and Stage representations of the path. It is used to control how the time span is distributed over the path on the Stage.

  1. Right click on path A (on the TimeLine) at around the 4.8 seconds position – slightly to the right of the newly added trigger event marker.
     Select "Add timesync" from the right click pop-up menu.
     Two violet markers appear – one roof shaped on path A on the TimeLine, and one tiny square one on path A on the Stage (see the image below).
  2. **Ctrl-drag** the square marker slightly to the left towards the edge of the moon – as indicated by the red arrow in the image below.
     Test run: The effect should be that the rocket moves towards the moon at a slower speed, and then speeds up when the rocket engines are ignited.



- Save the animation again, and feel free to continue adding new actors and/or paths and properties/events. An available and suitable alternative or complement to the rocket ship ("rocket.png"), is for example the "FlyingSaucer.png".

- A somewhat extended version of the animation we've created can be found in the "PassiveAnim" folder under the name "MoonCometX.omega_anim". Take a look and play around with it, and compare with the result of our above exercise.

Advanced functions

In addition to the presented functions, the Animation Editor supports a number of options that we haven't had the opportunity to utilise in the above examples. Please see the list of functions and commands below for information about these. But before that we wrap up with some words about internal animation of actors, the possibilities offered by using variables in the animations, and some tips and tricks.

Internal animation

As mentioned above (in ”Adding and adjusting new actors”) internal animation – i.e. the movements of the actor itself, e.g. a ”walking” motion – is accomplished by the program automatically flipping through images with the same name, but with the extension -01, -02, etc. up to -00, and with a default speed of 0.2 seconds per image. This has the following consequences:

- If you link to an actor image in a folder where there are more images with the same name plus additional -01, …, -00, then the actor will automatically be presented with an internal animation based on this series of images.

- The series must always begin with ”imagename-01.png”, and ended by ” imagename-00.png”. The more images in the series, the more detailed or complex internal animation can be performed. The speed of the internal animation is by default 0.2 seconds per image (5 images per second), but this may of course be adjusted in the Marker Properties for the actor’s TimeLine.

- If you want to avoid an internal animation you must refer to an actor with an image name that is not part of a numbered animation series, or by calling an attribute (of an animated image) that lacks an animated series – an example:

  In the C:\omega\media\actor\FeedEat folder we find many image series, including …
  Tiger.png, Tiger-01.png, Tiger-02.png, Tiger-00.png
  Tiger-Eating.png, Tiger-Eating-01.png,…, Tiger-Eating-04.png, Tiger-Eating-00.png
  Tiger-Wait.png, Tiger-Wait-01.png, Tiger-Wait-00.png
  Tiger-Still.png
  This means that by using the actor ”Tiger.png” we will have an internal animation based on 3 images, calling the attribute ”Eating” will give an animation based on a 5 series, with the attribute ”Wait” an animation of 2 images, but the attribute ”Still” will present a tiger without internal animation.

Using variables

Variables can be used in animations (as in lessons) to establish more advanced relations and links between actors and their paths, effect sounds and attributes, and to choices made in the lessons. The easiest way to illustrate this is to look at a couple of examples from animations in the Omega-is resources:

- **The simplified ”invisible” variable for effect sounds:**
  - Open the ”ChaseChased1.omega_anim” animation.
    This animation, like many others, shows a simplified way to call effect sound files using the current actors ID as a variable.
  - Open the Marker Properties dialogue for the first added Trigger marker (after the start marker, Ordinal=2), on path A on the TimeLine. There, the rather anonymous sound file ”audio/sound.wav” (or .mp3) is referenced in the Play Sound argument. This means that a sound file with the name ”sound-” plus the actor ID of the used actor will be played (if it exists) – e.g. ”sound-lion” with the lion actor. In this fairly simple way, relevant effect sounds may be linked up to actors used in the animations at each specific occasion.

- **Explicit variables for effect sounds:**
  - Open the ”StopsInRelationTo1.omega_anim” animation.
    In the Actor Properties we can see that the three first actors have one variable each – ”carStop”, ”mcStop” and ”busStop” respectively.
  - Open the Marker Properties dialogue for the start marker (Ordinal=1), on path A on the TimeLine.
    There, at Play Sound, a sound file called ”audio/${A:1}.wav” (or .mp3) is referenced.
    This means that a sound named as variable 1 for the current actor on path A will be played. Explicitly, this means that the name of the sound file here could be ”audio/carStop.wav(mp3)”, ”audio/mcStop.wav”, or ”audio/busStop.wav”.
  - The corresponding goes for the sounds played at path B.
- **Variables in image attributes:**
  - Open the ”KickToPerson.omega_anim” animation.
    Among the actors there is a ball and a bunch of balloons.
    As seen in the Actor Properties, each actor is represented by the images ”\actor\KickThrow\BallSoc.png” and ”\actor\KickThrow\balloon.png”. There we can also see that these actors each has a variable – ”Ball” and ”Balloons” respectively.
  - Now, open the Marker Properties dialogue for the last trigger marker (before the end marker – Ordinal=4), on path C on the TimeLine. There an image attribute is used called ”Catch${B:1}”. Explicitly this means that the attribute will be ”Catch” + the value of variable 1 for the actor on path B, i.e. ”Ball” or ”Balloons” according to above.
    So the image attribute will either be ”CatchBall” or ”CatchBalloons”, which matches the image files ”actor\KickThrow\BlondGirl3-CatchBall.png” and ”…\BlondGirl3-CatchBalloons.png”.
- **Links between variables in lessons and animations:**
  - Open the Lesson Editor, and then the lesson:
    ”…\lesson-en\active\J_Long_Sentences\J2\ J2.omega_lesson”.
  - Create the sentence ”The fast lemur chases the scared mouse across the meadow”.
    As seen in the information above the sentence line, the word ”mouse” links up to an actor with the Actor ID ”mouse:scared” on a path with Path ID ”B” in the animation called (via properties of the word ”chases” – including two variables).
  - Now, click on ”Edit animation”. The Animation Editor is opened with the ”ChaseFastOverMeadow.omega_anim” animation.
  - Open the Actor Properties e.g. for the scared mouse (3:d from the end). There you will find the Actor ID ”mouse:scared”, and this actor has also been equipped with the variable ”Scared”. This is used to play a suitable sound in path B on the TimeLine – see the Marker Properties for the first (”<”) marker (Ordinal=0).

- In this way the variable ”scared” of the word ”scared” in the lesson has been linked on to the variable ”Scared” in the animation, and there further on to a matching sound effect.
- From version 1.2 it is also possible from Marker Properties to refer directly to a variable of a word on a specific sentence label selected in the calling lesson; e.g. in Marker Properties - for Image Attribute or Sound file - type "${@S:2}" to pick up variable 2 from the word selected for label 'S' in the lesson/sentence that called the animation.

Tips and tricks

- Backup copies – prev and prevprev files: These backup copies are created each time an animation is edited and saved to disk. If a mistake is done, you can always revert to an earlier version by re-naming a ”animation.prev” file to ”animation.omega_anim”. Now and then it is a good idea to trash prev and prevprev file garbage in the animation file folders.
- If two actors need to be animated on closely aligned paths, it is a good idea to do as follows:
  - Create the path for one of the actors and refine it to contain all, or most of, the necessary details – both on stage and on the TimeLine.
  - Then use the ”Duplicate” path command to create an exact copy of the first path, and then make the desired adjustments for the copy: E.g. move the whole path using Shift+drag (with the mouse) – both on stage and on the TimeLine – to achieve a desired relation between the actors in space and time during the animation.
- Save your own newly created animation files in the folder:
  ”anim\MyAnim”
  and accompanying new media files in the corresponding media folders:
  ”media\background\MyBackgrounds”
  ”media\Wings\MyWings”
  ”media\actor\MyActors” and
  ”media\audio\MyAudio”
  In this way it will be easier to keep track of which files that need to be included, and where to put them, when your own new animations and lessons are copied to other computers – and hopefully also to the Omega-is web site repository at www.omega-is.com to be shared with others.

Functions and commands

Below is a list of functions and commands in the Omega-is Animation Editor:

File:

 **New** – opens a new and empty animation file

 **Open** – displays the open file dialogue - to select and open an existing animation file

 **Reopen** – opens the latest saved version of the current animation

 **Save** – saves the currently open animation under the current file name

 **Save as** – displays the save file dialogue – to name and save the current animation

Exit – quits the Animation Editor – after a control question and confirmation

Canvas:

Set background displays the open file dialogue - to open an existing background image

 **Left** – moves the whole stage, background and all, to the left on the screen

 **Right** – moves the whole stage, background and all, to the right on the screen

 **Up** – moves the whole stage, background and all, upwards on the screen

 **Down** – moves the whole stage, background and all, downwards on the screen

NOTE – the above movement commands may be handled by mouse drag & drop

 **Smaller** – zooms out the background/stage by around 25%

 **Bigger** – zooms in the background/stage by around 30%

 **1:1** – resets the size of the background/stage to the default scale

 **Fit in window** – zooms the background/stage to fit the available screen size

Path:

 **Create new** – adds a new actor path – on the stage and the timeline (up to 4 paths)

 **Duplicate** – adds a coy of the currently selected path on the stage (incl. its timeline)

 **Extend at end** – adds another segment at the end of the currently selected path

 **Split in two** – splits the segment to the left of the selected path point into two parts (the segment to the right of the start point will also be selected and split)

 **Delete segment** – deletes segment t.t.l. of the selected path point (or t.t.r. of start point)

 **Delete whole path and timeline** – does exactly so – after a control and confirmation

TimeLine (animation):

 **Play (Skift-P)** – runs/plays the animation

 **Stop (Skift-S)** – stops the playing

Cast:

Set actor – displays the dialogue to open an image file for actors 1-4 in the actors list – corresponds to **Image name – Set** in the **Actor Properties** dialogue – see below

Add a wing – displays the dialogue to open an image file for a wing/prop in the next free slot in the Wings list – corresponds to **Image name – Set** in the **Wings Properties** dialogue – see below

Show actor properties… (Ctrl-A) – displays the **Actor Properties** dialogue – corresponds to a mouse right click on a slot in the actors list, or a click on the "**Prop…**" button to the left of the actors/wings list

Show wings properties… (Ctrl-W) – displays the **Wings Properties** dialogue – corresponds to a mouse right click on a slot in the wings list, or a click on the "**Prop…**" button to the left of the actors/wings list

Flip actor/wings (Ctrl-F) – flips between the actors or wings list – corresponds to a mouse click on the "Actors" or "Wings" buttons t.t.l. of the actors/wings list

Help:

Show manual – opens the editing chapters of this manual in the default web browser

About Omega-is – displays an information panel about Omega-is

About Anim Editor– displays an information panel about the Animation Editor

Commands and functions at the TimeLine:

Scale – changes the scale of the TimeLine. The  button extends the scale range, the  button narrows it (zooms in – down to 0.2 of the default scale)

Speed – changes the speed of the animation. The  button increases it – up to 2.0 times the default speed. The button decreases it – down to 0.5 of the default speed. NOTE! – These changes are only temporary to test the animation behaviour at different speeds (which may be set per user), and are not saved in the animation file.

Prop… – this button displays the Marker Properties dialogue (for the start marker of the first/top path on the TimeLine). The dialogue will then show the properties of the marker that is focussed by a mouse click – see further below.

The Right Click menu on the TimeLine:

Marker properties – shows the Marker Properties dialogue with the data for the closest marker on the selected path on the TimeLine.

Add timesync – adds a timesync marker to the selected path on the TimeLine, plus a corresponding marker on the animation path on the stage.

Add trigger – adds an activity trigger marker the selected path on the TimeLine.

Cancel – closes the Right Click menu.

The Ctrl + Right Click menu on the TimeLine:

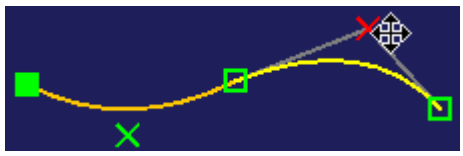
Marker properties – shows the Marker Properties dialogue with the data for the closest marker on the selected path on the TimeLine.

Marker delete – marks closest marker on the selected path on the TimeLine in red for deletion, and displays a Yes-No dialogue to confirm or cancel the action.

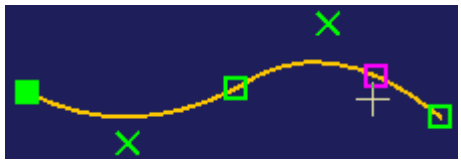
Cancel – closes the Ctrl + Right Click menu.

Direct manipulation (with the mouse) of paths and path points - on Stage:

Common path points – (start, segment, end and arc points) **click-drag** with the left mouse button to the desired position. (All points in the image t.t.r. – selected point is red – NOTE the movement mouse cursor!)



Timesync marker – Ctrl + click-drag with the left mouse button to the desired position. (Violet point t.t.r. – NOTE the cross mouse cursor!)



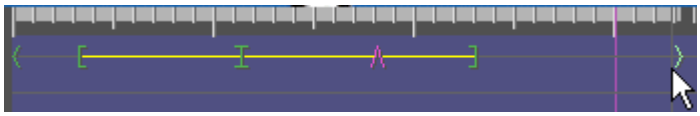
The whole animation path – Shift + click-drag (in the path point of your choice) with the left mouse button to the desired position.

Direct manipulation of paths and markers - on the TimeLine:

Positioning the path representation on the TimeLine – Shift + click-drag (at any point of the path) with the left mouse button to the desired position – alternatively: **Ctrl + click-drag** the path **start marker**.

Adjusting the length of the path on the TimeLine – Ctrl + click-drag the **stop marker** of the path to the desired position/length.

Adjusting the length of the animation time on the TimeLine – Ctrl + click-drag the **end marker** to the desired position on the TimeLine.



Trigger and timesync markers – Ctrl + click-drag the marker to the desired position on the TimeLine.

Image formats

Backgrounds:

- **JPG image files** – file suffix **.jpg** or **.jpeg** – are generally the natural choice, in particular for photo background images (millions of colours)
- **PNG image files** – file suffix **.png** (millions of colours or 256 colours)
- **GIF image files** – file suffix **.gif** (256 colours)

The 256colour alternatives may be suitable for drawn images with distinct colours.

Actors

For actor images it is necessary to have figures with transparent backgrounds, which can move nicely across the background. This motivates the following choice of file formats:

- **PNG image files** – either in 256 colours with indexed transparent background colour, or in millions of colours with a so called Alpha channel for the transparent background.
- **GIF image files** – 256 colours with an indexed transparent background colour. Animated GIF images (for internal animation) may also be used, but the internal Omega-is method for internal animation is recommended for a more predictable result.

Wings/props

The same image format as for the Actors above (except for animated GIF images).

Chapter G

Chapter G: Educational perspectives

MIR – about our method

The three letters M+I+R in many ways cover the working method proven to be the most successful. M stands for "Multimedia" and sums up the potential and possibilities provided by technology; I stands for "Interaction" and emphasizes the importance of an adult conversation partner being present as often as possible; R stands for "Recasting" and indicates the use of a special conversation strategy.

The computer as a linguistic prop

Since 1990, several studies conducted at Göteborg University, Sweden, have explored how the use of a multimedia based method may help learners with diverse linguistic disabilities develop their language skills (Tjus, 1998). The strategy we have used (the MIR method) is based upon three equally important elements.

1. **Multimedia**: In our research, we have operated with three different programs, Omega-is being the most recent and complete, although all programs have been founded on similar principles. These programs give the learner the means to create motivating and interesting events on the computer screen.

2. **Interaction**: Although the learner is sitting at a computer, interaction with the teacher is still a central feature. The goings-on on the screen provide the users with topics for conversation. The adult should, however, give as few cues as possible in favour of functioning as a helper and a source of inspiration to the learner.

3. **Recasting**: This is a well-known but underestimated strategy, which too seldom is used in a purposeful way. Several researchers (see Tjus, 1998 for a detailed review) have showed that learners learn new language concepts quicker if the adult recasts the learner's words. This is, in itself, a rather uncomplicated strategy: You change the linguistic style without distorting the essence of the learner's statements. Experience shows that there are more advantages to this technique than simply imitating the learner. One will maintain the same focus on the learner's utterances, one adjusts to the learner's pace and level, and the dialogue becomes more varied and genuine. Results from several studies give uniform evidence that a dialogue based upon these principles may have a distinctly positive effect on the development of language skills.

This model has been tested in several projects in Sweden during the 1990's with several groups of pupils (such as children with autism, multiple handicaps, and children with dyslexia). Results have, so far, been positive. Most learners - and this most certainly goes for the group of children with autism – have improved their reading skills, their communicative skills or their linguistic awareness (Heimann, Nelson, Tjus and Gillberg, 1995; Tjus et al., 2001). More broadly, research in the United States and Belgium as well as Sweden have demonstrated that Multimedia + Interaction + Recasting (MIR) approaches can support both literacy and basic language skills in ordinary children, deaf children, children with autism and children with multiple special needs.

Development of language skills and teacher – learner interaction

From today's research-based knowledge in education, psychology, and linguistics, it is possible to compile a number of conditions that strongly affect the normal language development of children (Heimann and Tjus, 1997). These conditions can, in basic terms, be divided into three groups: *necessary, facilitating,* and *fundamental* factors.

For each of these domains, examples of a few of the most important factors are given below. All *fundamental* factors contribute to the child's language development and will to communicate - a capability which takes an important step forward when the child at the end of the first year of life enters the world of spoken language. Regarding the *necessary* factors, they all have to be available for the language to develop normally, while, concerning the *facilitating* factors, the more of them that are present, the better!

Fundamental factors

Most of the fundamental factors usually being discussed have been taken from our knowledge of infant development. Today, we know that the embryo of our linguistic and communicative skills is present at birth. What takes place during the first year of life has, in other words, consequences for the linguistic development of the young learner. Early capabilities, supposed to be important to linguistic development are (adopted from Heimann and Tjus, 1997, pp. 56-57):

- Imitation. Infants imitate facial expressions (such as a happy or sad face) and facial gestures (such as mouth opening and tongue protrusion) already at birth
- Mother's voice. The infant recognizes its mother's voice even before birth. The infant can also recognize stories and rhymes that its mother has read to it during the last trimester
- Sucking. The infant's sucking pattern depends on how hungry it is. When an infant is lying at its mother's breast, most exhibit a sucking pattern of intensive sucking interspersed with pausing. A mother will

spontaneously and intuitively use this pattern to develop her baby's ability to take turns, often by simply talking to the infant during pauses.

- Turn-taking. Infants also seem to possess a completely innate ability to take turns in "talking" to the adult. If the mother or father only gives the baby time – infants need more time to answer – lengthy dialogues can develop. Such dialogues have also been observed between premature babies and their parents.
- Mother-infant interactions over time: It is important to the infant to have the possibility to continuously and consequently develop both its interaction and communication skills during its first year of life
- Eye contact. The fundamental ability to maintain eye contact is present already at birth, but it also quickly develops during the first months of life. Eye contact is an enormously important part of early contact and early interactions between infant and parents.
- A secure relationship. Our initial experience of the importance of communication and our initial experience of how important we are in the eyes of the world around us is given within our initial relationship. The foundation for this is built during the first year of life
- Joint attention. Around 8-9 months of age, one notices the first signs of the infant's ability for joint attention (as when the nine-month-old infant and its mother play with a toy and alternately look at each other and the toy). This ability is also connected to early communicative gestures (for example, pointing at something) and has proven to be a basic element for later linguistic and cognitive development

Necessary factors

(see Heimann & Tjus, 1997; Nelson, Loncke & Camarata, 1993)

- One competent partner. Sufficient dialogue episodes with one skilled partner (this person is typically an adult having a significant higher lingual competence than the learner).
- A sufficient numer of episodes. The learner must be given sufficient opportunities to interact and experience conversations with others whose language is both fluent and varied.
- Motivating conversations. Interactions and conversations that are both motivational and socially interesting to the learner, and where the learner is given the chance to participate according to his or her ability.
- Use of intact abilities. That the learner, in conversation, is given the opportunity to use biologically intact senses. For example, that profoundly deaf children are allowed to communicate through signing.
- Challenging conversations. That the learner at some points experiences a manner of communication that challenges the limits of his or her current linguistic system.

Facilitating factors

A few facilitating factors are exemplified below.

- Many modalities. Receiving language input and experience through as many modalities as possible (for instance speech, text and sign language). Simultaneously viewing a text, hearing it and also having it translated into sign language gives the learner more opportunities to understand and analyse relevant language content.
- Many challenges. Conversations that often challenge the learner's language comprehension and level of development give the learner extra stimulus and opportunities to discover new language structures as well as new ways of maintaining a conversation. One group of children at risk of not being taken seriously where language is concerned is multi-handicapped children who can perhaps express their own thoughts only through computers or a so-called BLISS-board (by means of using signs and symbols). This often leads to a slow conversation process, with short, clear-cut messages. Unfortunately, sometimes the obstacles caused by the motor handicap, together with the limitations presented by the computer or BLISS-board, cause others to believe that the child understands no more than he or she is able to produce.
- Flexible adjustment. That the person or persons who talk to the learner use several different means to make concepts or events understandable. This means that the adult modifies the speed as well as the complexity of the spoken language to suit the learner's present motivation, concentration and ability. It also includes the use of different means to make information redundant. This could mean describing the same event in several different ways, or using other modes of expression (such as drawing or singing) to increase the learner's ability to understand.
- Individual credit. The language spoken by the important adults surrounding the learner should be modified to suit the learner's personality, way of thinking (cognition) and social style. This is important because it acknowledges the learner as a separate individual with unique interests and needs. Even though much can be gained by studying literature on the subject of talking to children with autism, one must never forget that there is no one way to relate to children diagnosed with autism. These children, along with all normal children and all children with disabilities, are different and must be met as unique individuals.

Heimann and Tjus (1997) are of the opinion – as are Nelson, Loncke and Camarata (1993) - that if as many as possible of the factors listed above are present, the probability increases that the learner is placed in a favourable learning situation where the learner's brain/cognitive system is given a chance to absorb and analyse vital information. Thus, the learner has a better chance of actually learning something new, for instance a new linguistic form that he or she has not mastered before. According to Heimann and Tjus (pp. 60-61; authors transl.):

A child must not be viewed as a biological computer to be fed with certain pieces of information. The child is a whole human being with human needs! Thus, learning neither occurs exclusively at school nor happens in a situation where learning factors (such as aptitude, memory and attentiveness) can be viewed as disengaged from the social situation (the child's emotional life, interests, self-confidence or previous history).

L.E.A.R.N: Some more theory

Here is a description of a somewhat different way of looking at learning. This is a way, which in our mind, can contribute to new ways of thinking in matters concerning teaching set-up and procedure. The great advantage – in our opinion – with our theoretical take on learning is that this theory focuses not only on the learner's abilities or inabilities, but instead attempts a comprehensive view on learning

The theory upon which our reasoning is based – *rare event learning (REL)* – is geared towards identifying and more thoroughly describing factors that are either necessary for learning to take place, or in one way or another facilitates learning (Tjus, 1998; Nelson et al., 2001). These factors are not easily identifiable in the classroom, but an increased awareness about them will hopefully improve chances of creating a learning situation and a classroom climate that maximizes the learners' opportunities to absorb new information. This view may require emphasis to an even greater extent for children with different forms of learning disabilities, since they, in contrast to healthy children, have fewer intact psychological (such as memory, language capability, concept building) and biological (such as sight, hearing, sensory and motor control) systems to employ.

The theory brings up five fundamental aspects that are important to consider when creating a positive learning climate. All of them describe, in different ways, how the learner's learning is guided by his or her ability to think, analyse, emotionally regulate and concentrate. But - and this is important – this theory is more encompassing than this, since it involves more than the usual cognitive conditions by emphasizing the importance of motivation, interest, feelings and the interaction between learner and teacher. Learning always occurs in a larger context. Moreover, within context these multiple learning conditions need to dynamically co-occur and converge--thus the more recent labelling of the Rare Event Learning model as a model of Dynamic Tricky Mixes (Nelson, 2001; Nelson et al., 2001; Nelson, Craven, Xuan, and Arkenberg, 2004). It is complicated or "tricky" to find the "right" Dynamic Mix that leads to rapid learning for an individual. But when found these individualized learning mixes may lead to highly productive cycles across many learning occasions with increasingly positive engagement and enjoyment by both child and teacher.

In short, the main content of the REL model can be described by using the acronym L.E.A.R.N (Nelson et al., 2001; Tjus & Heimann, 2001):

- **L**aunching conditions include cognitive factors such as the learner's ability to attend to new and challenging structures that are experienced within a verbal dialogue. It has to do with how prepared the learner is to learn new things, how able he or she is to capture relevant information and how easily those new structures are encoded into the learner's long-term memory. In addition, the teacher may have too high or too low initial expectations of the learner's learning potential and thus create a setting in which the learner feels uncomfortable and/or unmotivated. It is usually difficult to decide what particular levels or zones of challenge

are needed to allow optimal processing and learning by a pupil at a particular stage of acquisition in speech and text.

- **E**nhancing conditions might include catalysts such as recasts: The teacher reformulates the children's utterances into new syntactic packages, keeps the central meaning but may add something, as in the following illustration

  Child: "There is a dog."
  Adult: "Yes, there is a dog and it is big, isn't it?"

- **A**djustment processes include factors such as support from the teacher to the students to overcome frustration and positive reinforcement of the children's self-esteem. This part of the teaching process is particularly important since children with learning disabilities are at high risk for developing associated emotional and adaptive disorders that might contribute to emotional obstacles within the learning process. Poorer self-concepts (general and academic) and lowered expectations have been noted as possible negative outcomes.

- **R**eadiness conditions are illustrated by the children's level of functioning and their interest, knowledge and motivation for learning. Thus, assessment of various cognitive abilities is needed before the aims of training are decided. This is extremely important for children with autism since it has been found that (1) communicative language and (2) cognitive levels are the factors that best predict future progress. In addition children with disabilities often have problems with self-regulatory strategies that might lead to a lack of checking, planning, and monitoring. They often have difficulties with detecting relevant details and they seldom modify their strategy even after several rounds of negative feedback.

- **N**etwork representations. When new language material becomes well rehearsed and well-integrated within existing knowledge, new and efficient representations are formed. As a consequence, allocation of processing energy becomes more sufficient. For example, a good reader must be capable of parallel processing since reading a story requires considerable energy to keep track of all the lexical, syntactic, and discourse information comprising a story. A poor reader usually spends too much time and energy on letter identification which often interferes seriously with higher skills such as comprehension of the text material. Multimedia materials of the type we have used support parallel processing of more than one structural comparison.

In addition to LEARN, there are a number of additional processes that take a central position in our understanding of the learning process of children (Nelson, 2001; Nelson et al., 2001, 2004). Some of these are briefly described below:

1. Engagement. The learner's cognitive system must be engaged in the on going learning process. Language learners continuously compare new linguistic forms with the language they already master. We are referring to mainly subconscious cognitive and linguistic processes. When the brain (the cognitive-linguistic system) observes a difference, the

discrepancy is examined by these subconscious processes. When the discrepancy has been observed and examined a sufficient number of times, the learner develops a new language skill and is from that moment on able to understand and use the new structure.

2. Memory. As new and unknown lingual expressions appear, these are compared to examples already stored in existing memory structures. The memory is, however, not like a tape recorder storing everything without any distinction. The learner primarily observes certain very obvious and easily identifiable examples.
3. Consolidation. When the learner possesses full mastery of a new skill, such as being able to use a new language structure without difficulty, his or her nervous system no longer needs to pay attention to it. In other words, the brain can at this point just let go of this specific linguistic form and instead begin to focus upon new forms that have not yet been completely mastered by the learner. This process is chiefly, if not completely, subconscious. The learner's interest in different matters also, in part, determines what is paid attention to.
4. Integration. The brains of children are actively looking for patterns and connections to tie different areas of knowledge together. The recently acquired knowledge is tied together with previous knowledge, and broad patterns of experience are created. This tendency towards integration is present in all human beings and works as an important prop to learning.

On the importance of conversations: recasting

Keith Nelson, an American researcher who has studied early language development since the 70's is also one of the co-founders of the Omega-is program. He stresses the fact that common conversation, the dialogue between teacher and learner, is a resource rather seldom used in regular teaching. Dialogue is all too often an act where the teacher administers orders or reprimands or in other ways is at the head of the interaction. Conversations too seldom focus for extended periods on what the learner actually says or direct his or her attention towards. Yes, even when the teacher actually is trying to get the learner's message, it all too often happens that the teacher answers too soon or on a too advanced language level.

How can one, as a teacher or a parent, reduce one's own pace, one's own demands on performance, and start listening to the learner instead? Two strategies often used in research studies as well as in educational praxis is to either imitate the learner or try to recast what he or she just said. These strategies are briefly described below, even though it is the recasting strategy that we primarily recommend when working with Omega-is (from Heimann and Tjus, 1997):

- *Imitation*. One language training strategy often used - especially in the US and in England – is to have the children imitate words and phrases

selected by the teacher as especially important in regard to the development of the children. But imitation may also be used to facilitate conversation and interaction between teacher and learner. In infant psychology, this has been obvious for a long time: parents who don't succeed in establishing eye contact with their baby can get a better understanding of the interaction by imitating the infant for a short wile, thereby giving the infant a better opportunity to participate. Imitating forces you to focus on your partner, to adjust to his or her pace, thereby giving the "weaker party" room to claim more space in the dialogue.

- *Recasting.* This is an undervalued strategy, too seldom used in a deliberate manner. The strategy is quite simple: The teacher changes the language form without altering the core message in what the learner has said. Studies show that this technique has more advantages than simply imitating the learner. One gets the same focus on what the learner has said, one adjusts to the learner's pace and level, and the dialogue becomes more varied and genuine. There is unequivocal evidence that a consistent teacher-learner dialogue of this type speeds up the pace of learning (Nelson et al., 2001).

Recasting - how to do it

We refer to Heimann and Tjus (1997) in the following brief description of how to work with the recasting method: The condition being that a conversation situation and dialogue is established around the work at hand (such as an Omega-is lesson). The teacher/parent tries to answer questions, ask, comment, expand and recast the dialogue. The dialogue is supposed to help the learner understand the meaning of the words and phrases he or she is currently working with:

- **Step 1.** It is important that the teacher or parent is sensitive to the learner's spontaneous comments and finds an unaffected manner in which to use the strategy (every remark from the learner cannot be recasted; if so, the method is at risk of becoming absurd). Let's assume that the teacher and the learner are discussing wild animals, watching an Omega-is lesson containing pictures and film clips of different animals and that the learner remarks "I like the big tiger".
- **Step 2.** The teacher must now make good use of the central content of the learner's remark and quickly recast it, giving it a partly different structure without changing the meaning to any large extent. One way of doing so might be saying, "Yes, it's a nice and big tiger" or "I like the tiger, too". An alternative to recasting is helping the learner elaborate by asking, "Why do you like it?"
- **Step 3.** As a final link, it may sometimes be convenient to follow up on the remark in step 2 by trying to direct the learner's attention towards the text shown on the computer screen. This can be done by asking something like, "Where's the word?" or by pointing out: "Here's the word! Look: Tiger".

Results from research

Many attempts at using computers in the education of children with language problems have been made since the early 1970s. In our own experiments we have worked with several different groups of children (such as children with cerebral palsy, hearing disabilities, ADD/ADHD), but the particular group that the larger part of our experience stems from consists of children with autism. The use of the computer as a teaching aid has been under much discussion in regard to this particular group; it has taken a long time for the computer to become an established device for these children. Today, several different studies show that the computer can be very useful to children with autism, as to children with other linguistic and communication disabilities. Children with autism can learn to read and count with the help of adapted computer programs, and computerized education can even make them learn more than they would do in traditional education directed at children with autism (Jordan and Powell, 1990; Tjus, Heimann and Nelson, 1998, 2001). Recently, two studies from Spain show similar positive results from a group of children with mixed developmental disabilities (Basil and Reyes, 2003; Reyes, Basil and Rosell, 2000).

The most extensive account of research results is to be found in Tjus' doctoral dissertation from 1998, but Tjus and Heimann (2001) also give a summary of results reached during their initial seven years of research. For additional information we recommend the attached bibliography and a visit to our web page www.omega-is.com.

Even though all of the groups of children included in our research have, for the most part, improved on their reading skills, the strongest and most obvious effects have generally been observed within the group of children with autism. In previous studies using the computer program preceding Omega-is (DeltaMessages; Nelson and Heimann, 1995), 52 children were included; 37 of these improved their reading skills. In other words, working with the computer had a positive effect on the reading skills of over 70 per cent of the participating pupils. For children with autism, a corresponding individual analysis shows that 10 out of 13 children (=77 per cent) improved their literacy skills with our method. It is worth noticing that the total duration of intervention was only between two and four months. During this period of time, the children worked twice a week with the program (every session being about 20 minutes long, and the teachers were instructed to use the described MIR strategy).

Increase in reading (Project Delta*Messages)*

(Tjus 1998; Tjus, Heimann & Nelson, 1998)

Participants: Children with:	YES	NO
Autism	10	3
Hearing Impairment	7	4
Dyslexia/Adhd	12	5
Mixed Intellectual Disabilities	8	4
	37	15

Selected Bibliography

Basil, C. & Reyes, S. (2003). Acquisition of literacy skills by children with severe disability. *Child language teaching and therapy*, 19 (1), 27-48.

Heimann, M., Nelson, K.E., Tjus, T., Gillberg, C. (1995). Increasing reading and communication skills in children with autism through an interactive multimedia computer program. *Journal of Autism and Developmental Disorders, 25* (5): 459-480.

Heimann, M., Tjus, T. (1997). *Datorer och barn med autism*. Stockholm: Natur och Kultur.

Jordan, R. & Powell, S. (1990). Improving thinking in autistic children using computer presented activities. *Communication*, 24 (March), 23-25.

Nelson, K.E. & Heimann, M. (1995). *DeltaMessages 2.0 - a multimedia software for language learning*. Distribution: Super Impact Images Inc., USA och Topic Data- och Språkbehandling HB, Göteborg. (© Keith Nelson & Mikael Heimann).

Nelson, K.E., Heimann, M. & Tjus, T. (1997). Theoretical and Applied Insights from Multimedia Facilitation of Communication Skills in Children with Autism, Deaf Children, and Children with Other Disabilities. In Lauren. B. Adamson & Mary Ann Romski (Eds.), *Communication and Language Acquisition: Discoveries from Atypical Development (pp.295-325)* Baltimore: Paul Brookes Publishers.

Nelson, K.E., Loncke, F. & Camarata, S. (1993). Implications of research on deaf and hearing children's language learning. In M. Marschark & D. Clark (Ed.). *Psychological perspectives on deafness*. Hilsdale, NJ: Erlbaum.

Nelson, K.E., Welsh, J., Camarata, S., Tjus, T. & Heimann, M. (2001) A Rare Event Transactional Model of Tricky Mix Conditions Contributing to Language Acquisition and Varied Communicative Delays. In K.E. Nelson, A. Koc, & C. Johnson (Eds.), *Children's Language, Volume 11*. Mahwah, NJ: Erlbaum.

Nelson, K. E. (2001). Dynamic tricky mix theory suggests multiple analyzed pathways (MAPS) as an intervention approach for children with autism and other language delays. In S. von Tetzchner & J. Clibbens (Eds.), Understanding the theoretical and methodological bases of augmentative and alternative communication (pp. 141-159). Toronto: International Society for Augmentative and Alternative Communication.

Nelson, K. E., Craven, P. L., Xuan, Y. Z., & Arkenberg, M. (2004). Acquiring art, spoken language, sign language, and text: Developmental and evolutionary observations from a Dynamic Tricky Mix theoretical perspective. In P. Bauer, J. Hudson, R. Fivush, & J. Lucariello (Eds.), The Mediated Mind: Essays in Honor of Katherine Nelson. Hillsdale, NJ: Erlbaum.

Nilheim, K., Heimann, M. & Tjus, T. (2002). *Datorn som pedagogiskt hjälpmedel för barn med autism: En attitydundersökning bland föräldrar och personal vid Rebeckaskolan.* Rapport från Psykologiska institutionen, Göteborgs Universitet, nr 1.

Reyes, S., Basil, C. & Rosell, C. (2000). Avalució del programa multimedia Delta per l'aprentatge de la lectura l'escriptura en alumnes amb discapacitas: una experiència d'ensenyament autoiniciat i motivador. *supports*, **4** (1), 51-65 [in Spanish].

Thorsell, A.M. (2002). *Datorn som pedagogiskt hjälpmedel vid läsinlärning – en studie fokuserad främst på barn med autism [The computer as a tool for reading support – a study on children with autism]*. Psykologexamensuppsats, Göteborgs universitet. [Master thesis, Dept. of Psychology, Göteborg University, Sweden]

Tjus, T. (1998). *Language and literacy acquisition in children with developmental and learning disabilities*. Doktorsavhandling, Psykologiska institutionen, Göteborgs universitet.

Tjus, T., Heimann, M. & Lundälv, M. (2003, Nov). *Multimedia enhancement of language and reading skills*. Paper presented at Autisme Europe, Lissabon.

Tjus, T. & Heimann, M. (2000) Language, multimedia and communication for children with autism– searching for the right combination. In S. Powell(Ed.), *Helping children with autism to learn (pp. 78-93)*. London: Fulton publishers.

Tjus, T., Heimann, M., Nelson, K.E. (1998). Gains in literacy through the use of a specially developed multimedia computer strategy: Positive findings from thirteen children with autism.(1998). *Autism , (2)*, 139-156.

Tjus, T., Heimann, M. & Nelson, K.E (2001). Interaction patterns between children and their teachers when using a specific multimedia and communication strategy: Observations from children with autism and mixed handicaps. *Autism, (5),* 175-188.

Tjus, T & Strid, K. (2000). Positive changes in syntactical reading errors in children with autism after multimedia training. Paper presented at the *XXVII International Congress of Psychology 23-28 juli 2000,* Stockholm, Sverige.

Omega-is has been developed and produced by the Omega-is Project Group:
Mikael Heimann**,** Mats Lundälv**,** Tomas Tjus and Keith E. Nelson

Programming:
Lars Nordberg – Femtio Procent Data AB, and Bengt Farre – Androtech HB
Animations
Mats Lundälv and Cathy Agostinelli
Graphics
Maria Olsson, Tommy Hagman, Oscar Elmgren, Lisa Wikberg, Marcus Anderberg, Daniel Sköld, Tomas Abrahamsson, Carina Fihn and Jill Lindström.
Sound work
Mats, Linnéa, Emma and Jakob Lundälv and Brian Holmedal
Languages
Katarina Mühlenbock (SE), Stine Laberg (NO), Elin Styve (NO), Rebecca Heimann (NO) and Mirja Turestedt (SE, UK)
Web page
Mattias Twedmark

Omega-is has been developed with support from
Sweden: The Handicap Institute, Vinnova/KFI, Center for Autism, Göteborg, Institute of Special Education, Göteborg, and DART/ The Queen Silvia Children's Hospital, Gothenburg
Norway: Center for Child & Adolescent Mental Health, University of Bergen and the Meltzer foundation, Bergen.

Information on how to order this program will be obtained from

Sweden:	Topic Dos Hb Barnhusgatan 13, SE 411 11 Göteborg, Sweden Email: topic@telia.com
Norway:	Center for Child & Adolescent Mental Health University of Bergen, P.O. Box 7800 NO-5020 Bergen, Norway Email: post@rbup.uib.no
USA:	Super Impact Images Inc. c/o Nelson, Warriors Mark, PA 16877 Email: Keith.Nelson@psu.edu