



UNIVERSITY OF MURCIA

*International Journal
of
English Studies*

IJES

www.um.es/engphil/ijes

Principles in Call Software Design and Implementation¹

ANA GIMENO SANZ*
Universidad Politécnica de Valencia

ABSTRACT

This paper discusses a number of principles to bear in mind when designing and implementing multimedia language learning materials. Issues such as personnel and tools are briefly looked at. The general programme design principles referred to are based on courseware features such as log-on, menus and hierarchy, reference materials, utilities, student recording devices, feedback, progress reports and exercises. A general overview is given on the template approach to designing and creating a multimedia CALL package. Throughout the paper examples are given from the author's own experience in multimedia development for language learning purposes.

KEYWORDS: inultiinedia, developinent, design principles, CALL.

* *Address for correspondence:* Ana M^a Gimeno Sanz, Departamento de Idiomas, Universidad Politécnica de Valencia. Camino de Vera. 14. 46022 Valencia. Tel: +34-963 877 530; Fax: +34-963 877 539; e-mail: agimeno@idm.upv.es

I. FIRST CONSIDERATIONS

Designing and creating a multimedia CALL package is an extremely demanding task, calling upon a range of skills and meticulous attention to detail. Such is the complexity of computer programs these days that it is highly unlikely that a single person will have all the necessary skills to undertake a CALL development project alone. Team work is therefore essential, and each member of the team must have some understanding of the roles of the other members of the team. The language teacher who joins a software development team, for example, does not have to possess computer programming skills but s/he must have some understanding of basic programming concepts. Similarly, those responsible for the programming do not have to have knowledge of foreign languages, but they need a good understanding of natural language processing.

Once the decision has been taken to create a multimedia CALL package it is essential to consider:

- The target group.
- The language level of the target group,
- The purpose of the package itself — whether it is for general or specific purposes,
- The methodological approach
- And the appropriate use of technology.

A needs analysis taking into account both the learners' needs and the needs of the teachers/tutors who will be using the software should also be carried out. The results of a detailed survey will help us determine, not only the language contents and level our target group/s is/are particularly interested in, but also the most appropriate pedagogical approach to follow and how we can offer solutions through technology to enhance their learning process.

II. PERSONNEL AND TOOLS

Besides the subject specialist(s), a CALL project development team must have at least one member who is capable of using an appropriate programming tool, e.g.

- A dedicated CALL authoring programme such as Neuro Concept's *Speaker* Duke University's *WinCALIS* or Lavac, etc.
- A multipurpose authoring package such as Asymetrix ToolBook, Macromedia AuthorWare, Macromedia Director, etc.
- A high-level programming language such as C++, Delphi or Visual Basic.

A language teacher can achieve good results with a dedicated CALL authoring programme, however the multipurpose authoring packages mentioned above require a

considerable degree of programming expertise in order to get the best results. A language teacher can achieve reasonable results with such packages if s/he is willing to persevere, but the time taken to learn a complex package such as Macromedia Director will probably outweigh the benefits. A professional programmer might feel more at home writing in a high-level programming language such as Visual Basic or C++, both of which the average language teacher would not consider using. Unless the language teacher decides to do it alone—which is inadvisable—a CALL development project requires, at the very least, close collaboration between language teacher and programmer.

An important factor to bear in mind also at an early stage is graphic design. A programme may be very well designed content-wise and it may run very smoothly, but it will fail to achieve its aim if careful graphic design is not taken into account. It is highly desirable to call upon the services of a qualified graphic designer to produce pictures and icons, and to give advice on fonts, colour, screen layout, etc. Such a person will be familiar with design packages such as Adobe Illustrator and Adobe Photoshop. Graphic designers often have a background in photography. If not, the services of a professional photographer may also be required—or, at the very least, a very good amateur photographer.

Finally, a sound engineer and video technician will be required if the package is to contain substantial amounts of sound and video. They should be familiar with software tools for manipulating sound and video.

III. GENERAL PROGRAMME DESIGN PRINCIPLES

III.1. Deciding on Courseware Features

Once we know the content requirements, language level and learning needs of our target group, we must decide what the general courseware features aim to be. We may be designing materials for completely autonomous learners, in which case we may intend to include as many tools as possible in order to support self-access learning (by providing reference materials such as grammar notes, explanations on functions of language, cultural context of the target language, a sound-enhanced dictionary or glossary, student assessment, notepad, etc.) or a more tutor-guided package in which, in addition to those mentioned above, we may think of including tools that will support the teacher such as a teacher's guide, detailed student assessment reports, tracking devices, etc.

One of our basic aims when designing a multimedia CALL package should be to provide the learner with all the necessary tools that will encourage language awareness and enable him or her to discover the intricacies of its use for themselves with the aid of the programme. To achieve this, it is crucial that the design specifications of the programme are as detailed as possible.

111.2. Log-on, Menus and Hierarchy

First we must consider what the function of the entry screens are going to be. For example, we may wish to include a log-on system that enables language tutors to identify their students in order to assess their work at a later stage during the course; we may wish to include several support languages, therefore its selection would also take place on entering the programme; or we may wish to include an on-line registration system that will allow the learners to communicate with their tutors via e-mail, etc. The log-on screens will naturally lead to a main menu screen to identify the way in which the course contents have been organised as well as being the gateway for the learner to choose the route s/he is going to follow throughout the course. It is always convenient to offer a default route to guide learners through the programme but without forgetting that one of the advantages of multimedia courseware is precisely its non-linear access to the materials. Whenever possible it is advisable that all the courseware contents be visible in order to avoid unnecessary frustrations as to how big or small the programme might be —whereas students and teachers are accustomed to flipping through text books and gain a good idea, with just a first glance, of what the book may contain, in multimedia software this is normally not true because most of the contents tend to be hidden. To illustrate two different ways of courseware content structure, let us have a look at the main menu screens of *Español Interactivo* and *Español en marcha*.

In *Español Interactivo*, as we can see below, the contents are organised into five modules (Fig. 1), each dealing with a given topic, and each of these is in turn divided into four study units (Fig. 2), each relating to a particular sub-topic. Having chosen a unit, the learner is offered the learning goals underlying that unit and is presented with the actual activity contents screen (Fig. 3).

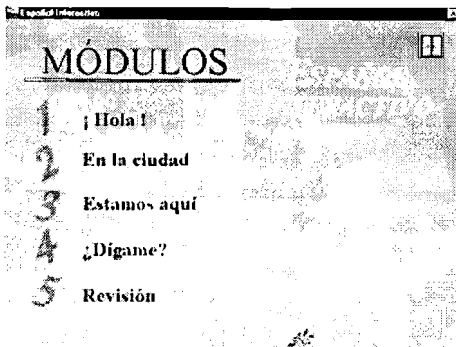


Figure 1: Modules



Figure 2: Units

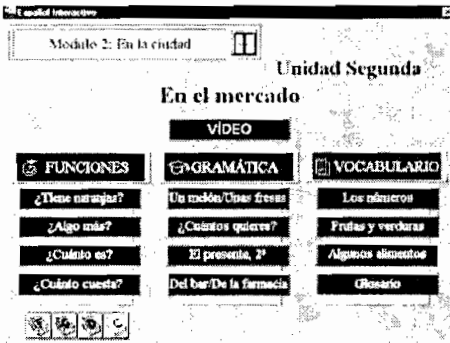
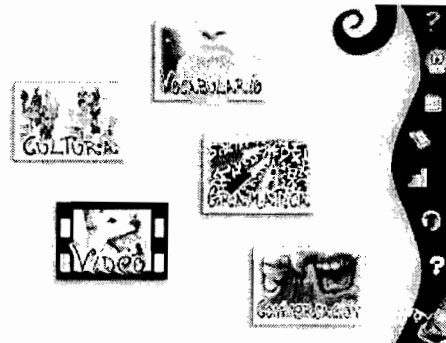


Figure 3: Activities

Figure 4: Main menu screen in *Español en marcha*

Every activity contents screen within the programme is arranged according to four main sections (video, language functions, grammar and vocabulary) and the activities within each section are related to the main topic underlying that study unit. Organised thus, it is intended that learners will follow the suggested progressive order and focus on the language components being practised in a given section. It is a means of dosing the language skills to be practised as well as a means of presenting language without stripping it from its context.

Another, perhaps more direct, way of presenting the contents to learners can be found in *Español en marcha*. Here, after selecting the support language, the main menu page displays five main sections —culture, video, vocabulary, grammar and conversation—, and the exercises are arranged according to their primary focus although in many cases they are interrelated. Each of these menu items gives access to a contents page listing all the activities. This method of arranging contents is generally preferred by teachers who intend to use the software as a complementary aid to their regular classes since it allows them to visualise the contents of the course in a more straightforward fashion than the previous example, which is more learner-centred.

When designing a multimedia language course we must always bear in mind what technology has to offer and how to make the most of it to the advantage of our students. Great care must be taken in drafting the global structure of the course taking into account all the links, resources, utilities, etc. we are going to include, all of which have to be consistent throughout the programme. The hierarchy of all the elements must be clearly structured to serve two basic aims; first, to allow the software programmer and graphics designer to visualise the entire programme, and second because it will also allow the language specialist to balance the contents, whether reference sources or actual activities. A common way to go about this is by creating a branching diagram specifying each of the elements integrating the hierarchy. The following is an example taken from *¡Bienvenido a bordo!*, composed of seven sections, each of these divided into a varying number of lessons —ranging from 4 to 11— and each lesson, in turn broken up into six activities. The total number of exercises amounts to over three hundred, therefore a hierarchical structure of the programme became crucial for the software programmer and the language specialist alike to bring all the elements together.

Sections	Lessons	Activities
1. <i>En el mostrador de facturación de equipaje</i>	1. <i>Sólo llevo equipaje de mano</i>	1. Dialogue
2. ...		2. Grammar Explanation
3. ...		3. Language Practice 1
...		4. Language Practice 2
7. ...		5. Vocabulary Practice
		6. Language Lab
	2. <i>Sí, viajamos juntos</i>	
	3. <i>Es imposible facturar hasta el destino</i>	
	...	
	7. <i>He perdido mi pasaporte</i>	
	8. <i>Llevo sobrepeso de equipaje</i>	

Table 1: Hierarchical structure of the programme

A hierarchical structure will also be indispensable to determine the programme's navigation system. Due to the intricate structure of CALL software, menus and sub-menus have to be clearly specified and back tracking options sometimes given so that our learners do not get lost in the programme. So this end a pull-down menu item or an icon should represent the following four actions: "back", to back track our steps; "next", to go to following item in chain of events; "menu", to return to previous menu in hierarchy; and "exit", to quit the programme.

11.3. Reference Materials

It is always advisable to allow access to all the reference materials at any point in the programme. Among the more common items normally included we might find: grammar explanations, use of language (or language functions), notes describing the cultural context of the target language, a sound-enhanced dictionary or glossary (which can of course be monolingual, bilingual or multilingual according to the needs of our target audience), etc. Linking these reference sources to the activities can be achieved very easily by including icons on the exercise screen that will call up a particular reference item. In the example below we can see a number of button icons representing four electronic reference books. These can be accessed either from within an exercise to respond to a particular question which may arise during the study process, in which case an additional window opens up instantly revealing the relevant explanation (which also saves time in having to look up an item in an index page), or they can be approached individually as self-contained sources in their own right and comprise their own navigation system by means of hypertext links. In writing these reference materials we should aim at including only what is relevant and appropriate for the course level and avoid superfluous explanations or going into a lengthy discourse that does not match the linguistic features exploited in the programme.



Figure 5: Grammar exercise taken from *Español Interactivo*

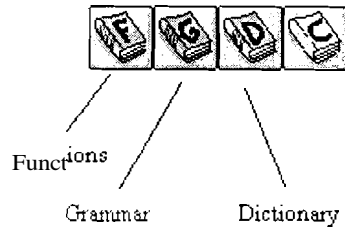


Figure 6: Detail of reference sources

11.4. Utilities

The utilities provided in the programme must also be consistent throughout. There are, of course, a large number of items that prove useful to learners. These can be presented either by means of icons discretely located on the screen or as a pull-down menu such as the ones found in any Windows-based programme. Among the more useful ones are:

- Help/instructions. We must distinguish between programme instructions —what the basic programic features are and how they operate—, help in the sense of giving a set of instructions to carry out an activity/exercise, and help in the sense of providing learning support and explanations referring to the target language. It is advisable for the latter two to appear as an additional item within the activity screen.
- A notepad to allow learners to take notes during their study process.
- A printer to enable learners to print a selection of text, such as grammar explanations, video/audio scripts, instructions, etc.
- Subtitles on/off to turn the script of a particular audio/video passage on or off. This utility enables us to create activities with two different levels of difficulty, i.e. an easier version of the exercise will be provided when subtitles are visible. We therefore recommend the default mode to appear with the subtitles turned off.
- Sound on/off to enable or disable all additional sounds which are not strictly necessary to complete an activity, i.e. audio instructions, positive/negative feedback, etc.
- Progress reports/student assessment reports to allow learners and/or tutors to check on the learner's performance in each exercise, session, etc.
- A recording device to allow learners to record their own voice and compare it with a model.

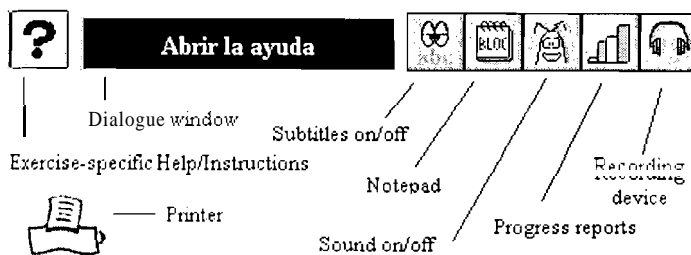


Figure 7: Icons representing program utilities in *Español Interactivo*

Although it is not advisable to use an excessive number of button icons, which might end up overcrowding the screen, we do recommend that however many icons are finally used, they should always appear in the same position throughout the programme. The graphical user interface must be as simple and as intuitive as possible at all times.

11.5. Student Recordings, Audio and Video

It is always beneficial to provide some sort of recording device that will allow learners to record their own utterances and compare them with pre-recorded models or provide graphical representations of their language input wherever possible. Some of the more entertaining activities are those where students can insert their own recorded versions of a dialogue, for example, and listen back to their own version in an attempt to simulate “real life” communication.

Interactive exercises of this type can be based on audio fragments, accompanied by illustrations or pictures, or complete video sequences. They can focus on pronunciation practice at word, sentence or discourse level. Creating exercises based on dialogues and requesting students to play the role of one of the characters is, to this day, one of the closest ways we can get to authentic communication in a CALL programme. These exercises are also very appropriate for pair work or collaborative learning—we can request two fellow students to record themselves and recreate a given dialogue completely, thereby doing away with the solitary task of studying with a language programme on one's own. Simulations between more than two or three characters are unadvisable to avoid interference on language retention.

Below are two examples taken from *Valencià Interactiu*. Fig. 8 illustrates the recording device linked to all the interactive exercises and Fig. 9 shows the video recording “studio” linked to video simulation activities. The utterances recorded by students can be stored, for the learner's future reference or for the tutor's supervision, in the temporary file directory although tutors must be reminded to delete these files now and then in order to save space on local hard drives.

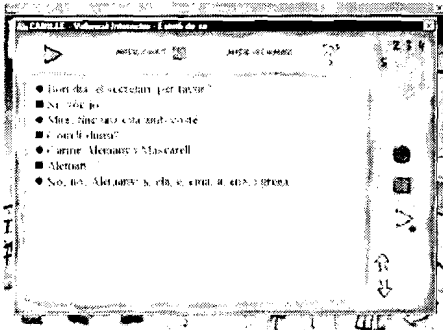


Figure 8: Activity audio recording utility

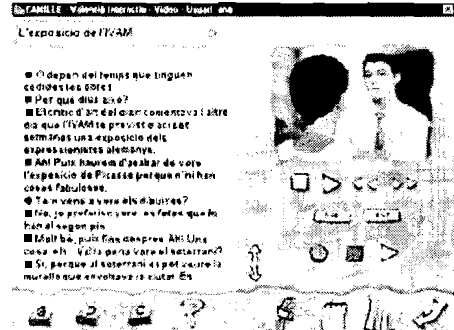


Figure 9: Video synchronised audio recording utility

Any student recording device in a CALL package should include such functions as:

- Play: to play current audiolvideo sequence
- Pause: to pause current audio/video sequence
- Instant replay: to instantly play current audio/video sequence from the beginning
- Fast forward: to play current audiolvideo passage fast forward and enable search
- Record: to record learner's voice
- Replay recording: to replay learner's last recording or sequence of recordings
- Compare: to compare learner's recording with pre-recorded model
- Subtitles/script: to show subtitles or full script of current audio/video passage

Since the storage capacity of CD-ROMs (or DVDs in the near future) can handle large amounts of audio, we must always take advantage of this media and include as much sound as possible in our courseware. It is no doubt beneficial to expose learners to as much authentic language as possible. Audio can be included in innumerable ways with or without text reinforcement—to give instructions, feedback, glossaries, exercise input, etc.—and will enhance the programme features considerably. Small amounts of audio can be recorded digitally using a high quality sound card—e.g. SoundBlaster—on our personal computer, edited with one of the many audio editing tools available—e.g. *Cool Edit*—and stored as an audio file in Windows PCM waveform format (.wav), for instance. The normal audio specifications for reproduction on a standard PC would be: a sample rate of 11025 Khz, mono, and a 16-bit resolution.

An important factor concerning audio is, of course, the quality of the sound recordings. We would recommend recording large amounts of audio at a professional studio with professional actors and actresses—which can actually work out cheaper than working with amateurs who may have to do numerous retakes over several days. When writing the materials, remember to keep careful track of what has to be spoken by a female or male voice—this can save a lot of time. The cost of video production can in many cases be decreased by using *chroma key*² background settings in a studio rather than filming on location. This also has the advantage of allowing more control over the sound quality and avoiding unwanted noise from cropping in. Video can sometimes be replaced by using a sequence of

synchronised still pictures and audio. The effect for language learning purposes can be remarkable and is considerably cheaper.

III.6. Feedback

Since one of the advantages of multimedia technology is the computer's immediate response to a mere touch of a key or mouse click, this is very useful when dealing with positive or negative feedback in reaction to the learner's performance in completing an activity or exercise. Learners, we have observed, tend to find it encouraging to read or hear immediate positive feedback when they have completed an exercise successfully.

A considerable variety of positive or negative text, audio or text and audio messages should appear/be heard at random and should be graded according to learner achievements. In a number of exercises feedback can be programmed depending on the number of attempts and a specific score given to each of these. We should include appropriate feedback for a "correct answer", a "partially correct answer", an "incorrect answer", or even exercise-specific feedback when a combination of options are required in order to complete an exercise successfully. An adequate way of offering positive feedback, for example, could be to have the audio file of a particular exercise play when a correct answer is given. Negative feedback, of course, has to be meaningful. It should always be clear what kind of mistake has been made and the feedback should provide not only awareness as to where the mistake lies, but also how to improve the learner's performance. Wherever possible we must avoid abrupt statements such as "No", "Incorrect, try again", but instead provide constructive criticism and try to anticipate and predict our learners' behaviour when completing an activity. This may be achieved by carrying out —prior to the design stage— an error analysis based, for instance, on L1 interference.

III.7. Progress Reports

Simple activity-based progress reports or more elaborate student assessment reports are of great value for both autonomous learners as well as for tutors intending to supervise their students' work. A simple scoring device indicating the number of correct answers out of a total can become a challenge that some students find motivating. In addition to the scoring device, we may choose to include a chain of score-dependent comments also aiming to encourage the learner to continue progressing.

SESION	BLOQUE	ACTIVIDAD	TIPO	FECHA	HORA	TOTAL A./I.	% TIEMPO
1	Gram.	Gramática	Activo	02/08/2002	12:30	24	80.00%
2	Gram.	Gramática	Activo	02/08/2002	13:00	24	80.00%
3	Gram.	Gramática	Activo	02/08/2002	13:30	24	80.00%
4	Gram.	Gramática	Activo	02/08/2002	14:00	24	80.00%
5	Gram.	Gramática	Activo	02/08/2002	14:30	24	80.00%

RESUMEN

SESION: 1
 ACTIVIDADES: 1
 PRECISION: 100%
 TIEMPO: 00:14:30

Figure 10. Student assessment—session report

RESUMEN

SESION: 4 BLOQUE: Vocabulario TIPO: Activo

ACTIVIDAD: ¿A qué hora empieza? (27 preguntas)

TOTALS: 18 ACERTOS/INTENTOS: 18 / 36 PRECISION: 50%

FECHA: 02/08/02 HORA: 16:40 TIEMPO: 00:01:37

COMENTARIOS

You're doing very well. Keep up the good work!

One answer was not answered correctly.
 Two answers were not answered correctly.
 Two answers were not answered correctly.

Figure 11. Student assessment—activity report

Figure 10 above is an example student assessment report taken from *Español en marcha*. The report summarises:

- The number of times learner x has logged on to the programme
- The amount of activities carried out per session
- Total time spent using programme
- Date and time of first session
- Date and time of last session
- Statistics showing average percentage of correct and incorrect answers per session and a table indicating, for every session, the names of the activities carried out, the section they belong to (grammar, vocabulary, etc.), the type of activity (passive/observation or active/production), the date, the current time, the total number of performances required by the learner to complete the activity, the number of correct answers in relation to the total number of attempts, the percentage of correct performance, and the time spent on the activity.

Figure 11, also taken from *Español en marcha*, illustrates a more detailed activity-based report. In addition to the information described above, by selecting a given activity, the learner or the tutor can further scrutinise the results achieved in a given exercise and instantly see where the difficulties may lie. For the benefit of autonomous learners primarily, a link from a given activity report to that particular activity would no doubt be greatly appreciated since it would save time in having to search for it through the content menus.

Since students, we have observed, are very keen on testing themselves and enjoy the challenges of being given a time limit, another useful way to monitor learner performance is to create activities that are time dependent. These activities are especially suitable for exam type entry and post-level achievement tests. A choice of time limits can also be given for a set number of questions in order to provide different levels of difficulty.

Figure 12 (below) shows a time dependent activity taken from *City Talk*. Displayed in the lesson plan (bottom right hand corner of the screen) are the five options available under the heading Quiz in the Reference pull-down menu. The options are: a 90 question quiz, random 30 questions, random 50 questions, timed 30 questions (25 seconds per question), and

lastly, timed 50 questions (15 seconds per question). All the questions are related to the grammatical, functional and notional contents practised throughout the programme. All five options provide an achievement scale for the learner to determine how well his or her performance has been in carrying out the test. All the reference materials, as well as the dictionary, remain available.

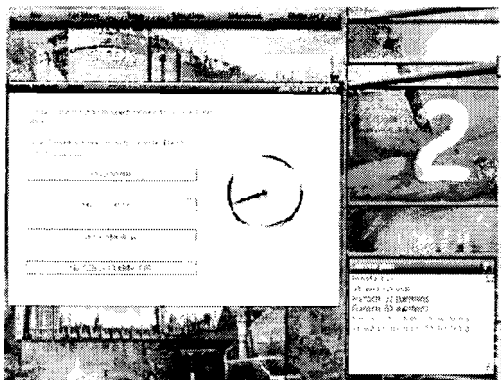


Figure 12: Time dependent activity

III.8. Exercises

When designing exercises for a CALL programme we must ensure variety, coherence, and consistency. Our materials must be based on sound pedagogical principles and firmly grounded on the methodology/ies we have chosen to follow. An attractive graphical user interface will enhance these qualities. We should try to avoid simply transforming typically textbook-type exercises onto a screen and instead try to be as creative as technology (and the software programmer) will allow us. Because technology moves so fast it is convenient to look ahead towards the future and try to envisage what the new tendencies might bring about in computer assisted learning.

One thing to bear in mind when writing exercises is that many of the more traditional activities such as multiple-choice and gap-filling exercises can be designed to function according to several modalities. For instance, a gap-filling exercise could also be designed as a multiple-choice exercise if the learner has to fill in a blank space from a list of given options, or as a drag-and-drop exercise if the learner has to select an item from a given list and drag it into a blank space. It is therefore advisable to combine as many different modalities of a sole exercise type in order to avoid repetitions that could perhaps result in our learners' boredom.

To put it schematically and focusing only on how an activity operates, a simple gap-filling exercise at word level in a given sentence, for example, could appear on screen in any one of the following ways:

1. Recorded audio input. Learner typewrites a whole word in a blank space. Learner hits enter key to check answer. If correct, audio recording is heard (positive feedback). If incorrect, appropriate negative feedback.
2. Learner typewrites a whole word in a blank space, choosing from a given list. If correct, audio recording is heard (positive feedback). If incorrect, space goes to blank again. Negative feedback.
3. Out of a number of options, learner drags and drops choice into blank space. If correct, audio recording is heard (positive feedback). If incorrect, space goes to blank again. Negative feedback.
4. Multiple choice: learner clicks on appropriate word from a given list. If correct, word appears in sentence and audio is heard (positive feedback). If incorrect, negative feedback.
5. Learner clicks on blank space and a window displays a number of options. Learner moves pointer to select option. Selected word fills blank space. If correct, audio is heard (positive feedback). If incorrect, negative feedback.

In "open input" type exercises where learners are requested to actually type in text (1 & 2 above), rather than choosing from a number of options (3, 4 & 5 above), it is always convenient to allow learners to make mistakes as long as we provide appropriate feedback and encourage a further attempt—a process similar to that found in spell checkers where incorrect/unknown words are highlighted but not automatically corrected for us.

Although the above modalities belong to a sole exercise type—filling in a blank space—, when they are exploited in a programme, a learner is under the impression that the exercises are completely different. Using a combination of these would therefore contribute towards gaining variety and diversity in our materials. The choice of one or another modality will of course depend on the primary objective of the activity itself. We should always ask ourselves what the learner will gain in terms of language acquisition. If we cannot find a substantial answer to this question, then the activity must be reconsidered.

Consider, for example, modality 1 above. This type would be appropriate if it comes after an "observation activity", a passive activity in the sense that we request the learner to observe, for instance a grammatical structure (e.g. comparing both forms of the verb "to be" in Spanish, *ser* and *estar*), and then ask him/her to participate more actively, to further explore the use and meaning of language and write the appropriate form of either verbs, providing perhaps a hint, but no options to choose from. Here we are asking the learner to actually write, allowing him/her to make mistakes, to actually put some effort into memorising all the possible verb forms. If the learner has to write the word in response to an audio input, s/he will additionally have to listen, interpret and then react (write the correct verb form). This modality is better suited for more advanced language practice.

In examples 2 to 5 above, there is no initial audio input. Instead the audio sequence becomes the positive feedback when the correct answer has been given. This does not mean that there is no stimulus for the learner to react to. We could obviously include sound or illustrations to prompt our learner. The difference, however, between modality 2 and

modalities 3, 4 and 5 lies in the fact that in the former the learner again has to write (and there is therefore more at stake), whereas in the other three a mere choice of option is requested. These modalities are better suited to lower level language practice.

The many combinations of elements available in multimedia —text, audio, graphics and video allow writers to create appropriate exercises/activities that suit a particular learning need. Computer assisted language learning is particularly suited to a communicative approach to language acquisition. Language can be presented both in terms of its structures (grammar, vocabulary, etc.) as well as in terms of its communicative functions; thus, our students will be able to learn linguistic structures, establishing relationships between structural and communicative functions in order to develop communication strategies in the target language. The functional and structural content of the course can be designed to encourage a range of communicative skills, bearing in mind that these are carried out within a specific social and cultural context.

IV. THE TEMPLATE APPROACH TO AUTHORIZING

This section looks at the template approach to designing and creating a multimedia CALL package. In the template approach to software development, content and functionality are distinguished so that the subject specialist —known in technical jargon as the content provider— and the programmer can work independently for a large part of the development process.

The task of the content provider in a CALL development project is not just to write texts in the target language. His/her job includes a range of tasks: e.g. besides writing texts, to write items for a range of different exercise types, to specify pictures that should accompany texts and exercises, to write scripts for audio and video recordings, to specify how the learner should progress through the programme, etc. The task of the programmer is to make everything happen according to the wishes of the content provider. It is essential that the content provider spells out in precise detail what is to appear on screen and what should happen as the learner interacts with the programme, in other words, a full description of what is known in technical jargon as an *event*. The programmer can advise on the feasibility of certain events, indicating to what extent the content provider's wishes can be carried out within the budget and time constraints of the project. The content provider and programmer have to agree on the exercise types, reference materials and overall functionality of the software before the main design phase actually starts. To this end a set of templates have to be worked out, into which the content provider slots the texts, exercise items, feedback, help notes, etc., and clarifies to the programmer how the different building blocks of each event are to be assembled³. A template looks rather like a form that has to be filled in, providing the programmer with the essential information that he/she needs in order to assemble the building blocks —known in technical jargon as assets. The assets of a CALL package might consist of:

- texts
- exercises
- photographs
- graphics
- audio recordings
- video recordings
- button icons
- pull-down menus

These assets are normally created and held outside the programme itself. For example, a word-processor might be used to write texts and exercise items, and photographs might be taken with a digital camera, stored in JPG format and manipulated with a graphic design package. The advantage of holding the assets outside the programme is that they can easily be altered without the intervention of the programmer. The content provider may, for example, find a number of typing errors in a text. These can be corrected in a word-processor and then the revised text file can be handed over to the programmer for incorporation into the programme. Similarly, the graphic designer may need to crop or retouch a photograph, or even replace it with a completely different photograph for a particular event—all of which can be done without constantly referring to the programmer. The task of producing different language versions is also made easier, as translators can simply create a new set of text files in the required new target languages.

A system of labelling all the events and assets has to be agreed, each event and asset being given a unique label, i.e. a name and/or number, so that it is clear what is being referred to. As an example, a labelling system could perhaps consist of:

- A letter to indicate the target language (which is useful when working with a team of colleagues creating several language versions).
- Five digits: the first one, running from 1 to 9, indicating Section/Module/..., the next two indicating Lesson/Unit/... and the last two indicating Activity/Exercise/..., each running from 01 to 99.
- Two digits to number pictures, photographs, graphics, video, etc. running from 01 to 99.
- The appropriate extension, i.e. .txt, .doc, .bmp, .wav, .avi, etc.

which would make each label look something like E5010310.wav, meaning this is the tenth audio file in exercise 3 of lesson 1 in module 5, in an English language course. It is always advisable to keep the labelling system down to 8 characters to avoid incompatibilities with some software programmes.

In the course of the initial process of consultation, the content provider and programmer will draw up a list of activities and exercise types. These may include:

- Multiple-choice exercises. Virtually all CALL packages include multiple-choice exercises —which may take a variety of forms: single-selection (when only one option is correct), multiple-selection (when more than one option is correct), select by number, select by clicking on icon, etc. In fact the combinations of question and answer can be: audio-audio, audio-image, audio-text, text-audio, text-image, text-text, image-audio, image-text. This exercise type is especially appropriate for checking reading or listening comprehension.
- Gap-filling exercises. Virtually all CALL packages include gap-filling exercises — which may take a variety of forms: open input (where learner types text), select from set of possible answers, drag-and-drop, etc.
- Language lab and prompted speaking activities. This type of exercise became possible with the advent of the MPC and now features in most CALL packages. Language lab activities are useful when we would like students to record their own voice and compare it with a pre-recorded model. The recorded model may range from isolated words, sentences or larger units of meaning such as exchanges in a dialogue. An interesting type of prompted speaking activity combines video and student voice recording.
- Clickable text on screen. This is a useful feature of CALL packages, allowing the student to call up audio, text or image files simply by clicking on words and phrases on screen. It can be very useful, for example, to provide reinforcement grammatical notes or to illustrate use of language within the broader context of a specific exercise.
- Clickable image on screen. This exercise type may be used when students are requested to click on a number of "hot spots" on a picture or image or click on a number of picture icons, in reaction to a written or audio input. For example, an activity to practice locations, where a map is displayed, and after hearing an audio input, the learner has to click on a specific place/item on the map. A less active modality of this exercise type is useful for introducing new vocabulary —having learners click on a picture or a selected area of a larger image with the effect of an audio fragment being heard and, optionally, its written version being displayed.
- Matching exercises. In this typology learners have to match two lists of items in any of the following combinations: text-text, text-image, text-audio, image-text, image-audio, image-image, audio-text, audio-image or audio-audio. The association of elements can be carried out by moving the second list of items to match the first list or, alternatively, both lists may remain static on the screen and matched by clicking consecutively on one and another item and indicating correct match with, for example, arrows or simply playing an appropriate audio sequence. The matches can of course be 1 to 1 (single-selection) or 1 to more than one (multiple-selection). The above combinations provide a considerable number of variations so as to allow writers to create a fair amount of entertaining as well as didactically efficient activities.
- Drag and drop exercises. A key feature of many CALL packages. Practically any type of exercise can be presented as a "drag and drop" activity: gaps can be filled by dragging text into blank spaces, matches/associations can be made by dragging one

itern (text, picture, etc.) over another. Yet there are a number of activities that do not fall into any of these categories, for example classifying exercises, which can be suitably solved by dragging items from one place to another over the screen.

- Reordering exercises. Also a key feature of many CALL packages. Reordering, or putting a number of items in their correct sequence, can take place vertically —reordering a dialogue, for instance—, or horizontally —reordering words/expressions in a sentence, for instance. Items to be reordered or placed in a specified sequence could also be pictures or audio sequences.
- Open-input exercises. Since it is very difficult to anticipate student errors and the number of programming variables involved so high, open-input exercises are useful, for example, in activities where the learner is requested to write —whether on the computer or not—, such as in free writing exercises where a model would be provided rather than a correct answer. It is also suitable for dictation or translation activities.
- Branching dialogues. These are a common feature in many CALL packages. They consist of dialogues in which a character on screen interacts with the language learner, who plays a role in a dialogue or series of dialogues that can branch in different directions depending on the learner's responses. The learner chooses from a set of given responses —since open-input from the learner would obviously be difficult to analyse. Branching dialogues can be based on a video sequence or on an audio file with accompanying still pictures. When designing a branching dialogue, it is useful to visualise the course of the dialogue by means of a diagram where we can also indicate the score given to each of the options provided.

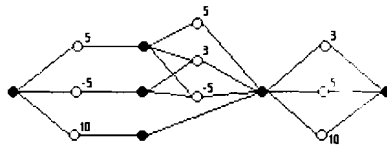


Figure 13: Diagram illustrating branching exercise. The numbers represent scores

The above list of templates is only a small selection of the many templates that language specialists and computer programmers might use in order to design a CALL package. Computing today allows well over thirty possible combinations of elements (audio, text, graphics and video) in a CALL program. By making discrete use of this huge choice it is possible to create attractive exercises and activities that enhance the interactive quality of the final product, avoiding repetitive and monotonous exercises that decrease students' motivation. All the elements that comprise a multimedia package are interconnected, so the templates must contemplate every single feature that appears on the screen at a given moment, e.g. links to and from exercises to reference materials, media to be included, etc.

NOTES

1. The findings outlined here comprise some of the ideas conveyed in *Information and Communications Technologies for Language Teachers* (www.ict4lt.org) and further explored in *CALL Software Design and Implementation: the template approach*, Valencia, SP-UPV, 2002.
2. A technique commonly used in TV by means of which an image is inserted into another, achieving the effect of a solid image. For example, recording a live person speaking with a photographic background. It is possible to achieve a very realistic effect suitable for language learning purposes.
3. However, this approach can be as simple as designing a series of word documents or as complex as producing an authoring shell.

BIBLIOGRAPHY AND REFERENCES

Printed Publications

- Blin F., Chénik N. & Thompson J. (Eds.) (1998). *CALL courseware development: a handbook*. Hull: EUROCALL, CTI Centre for Modern Languages, University of Hull. Available in Acrobat PDF format at:
<http://www.eurocall.org/courseware.htm>
- Chapelle C. A. (1998). Multimedia CALL: lessons to be learned from research on instructed SLA, *Language Learning and Technology Journal* 2, 1: 22-34. Available at:
<http://llt.msu.edu/vol2num1/article1/index.html>
- Collis B. (1995). *Telelearning in a digital world*. International Thomson Press.
- Davies G., Hickman P. & Hewer S. (1994). *Style guidelines for developers*. Hull: The TELL Consortium. University of Hull.
- Gimeno A. (1998). Developing multimedia CALL materials: *Español Interactivo*. In K. Cameron (Ed.) *Multimedia CALL: Theory and Practice*. Exeter: Elm Bank Publications. 289-301.
- Gimeno A. (1999). El aprendizaje de español/LE asistido por ordenador. CAMILLE: español en marcha. In Proceedings of the IX Congreso Internacional de ASELE, *Español como lengua extranjera: enfoque comunicativo y gramática*, Universidad de Santiago de Compostela. 837-846.
- Hoven D. (1997). Instructional design for multimedia: towards a learner centered CELL (Computer Enhanced Language Learning) model. In Murphy-Judy K. (Ed.) *NEXUS: The convergence of language teaching and research using technology*. Durham, NC: CALICO.
- Hubbard P. (1987). Language teaching approaches, the evaluation of CALL software, and design implications. In Smith W. (Ed.) *Modern media in foreign language education: theory and implementation*. Lincolnwood, IL: National Textbook Company.
- Hubbard P. (1992). A methodological framework for CALL courseware development. In Pennington M. C. & Stevens V. (Eds.) *Computers in applied linguistics*. Clevedon: Multilingual Matters.
- Kearsley G. (2000). *Learning and teaching in cyberspace*:
<http://home.sprynet.com/~gkearsley/cyber.htm>
- Krüger A. & Hamilton S. (1997). Individual language tutoring through intelligent error diagnosis, *ReCALL* 9.2: 51-58.

- Laurillard D. (1996). *Formative evaluation report*. Hull: The TELL Consortium, University of Hull. Available as a downloadable RTF file from: <http://www.hull.ac.uk/cti/eval.htm>
- Levy M. (1998). Two conceptions of learning aid their implications for CALL at the tertiary level, *ReCALL* 10, 1: 86-94.
- McCarthy B. (1998). Choice and constraint in software design, *ON-CALL* 8, 2. Available at: <http://www.cltr.uq.edu.au/oncall/mccarthy82.html>
- Myles S. (1998). The language learner and the software designer: a marriage of true minds or never the twain shall meet?, *ReCALL* 10, 1: 38-45.
- Plass S. (1998). Design and evaluation of the user interface of foreign language multimedia software: a cognitive approach, *Language Learning and Technology Journal* 2, 1: 35-45. Available at: <http://polyglot.cal.msu.edu/llt/vol2num1/article2/index.html>
- Riley F. (1995). *Developing multimedia courseware*. Hull: University of Hull. A downloadable book in RTF format: http://www.hull.ac.uk/itti/bk_dnl.html
- Salaberry M.R. (1906). A theoretical foundation for the development of pedagogical tasks in computer mediated communication, *CALICO Journal*, 14.1: 5-34.

CD-ROMs

- Gimeno A. (2001). *¡Bienvenido a bordo!*. Loidoi: Unicorn Training Partnership. Spanish courseware for the airline industry with English, Danish and German support languages, created with the Airline Talk Project. <http://www.airline-talk.com>
- Gimeno A. & Navarro C. (1998). CAMILLE: *Español Interactivo*. Barcelona: Difusión. Beginners Spanish multimedia courseware on 2 CD-ROMs + User guide. English and French support languages. Developed as part of the CAMILLE Project. Description in Spanish: <http://www.upv.es/camille> Description in English: <http://www.interactivespanish.com>
- Gimeno A. & Navarro C. (2008). CAMILLE: *español en marcha*. Barcelona: Difusión. Intermediate level Spanish multimedia courseware on 1 CD-ROM + User guide. English, French and German support languages. Developed as part of the CAMILLE Project. Description in Spanish: <http://www.upv.es/camille> Description in English: <http://www.interactivespanish.com>
- Gimeno A., Navarro C. & Montesinos A. (2000). *Valencià Interactiu*. Valencia: Bromera. Beginners course on 1 CD-ROM for learners of Valencian. Monolingual. <http://www.upv.es/camille>
- Gimeno A. & Navarro C. (2000). *Vida Urbana*. Loidoi: Libra Multimedia. Advanced Spanish courseware developed with the City Talk Project. <http://www.libra.co.uk>
- Limon D. & Mishan F. (2000). *City Talk*. Loidoi: Libra Multimedia. Advanced English courseware developed with the City Talk Project. <http://www.libra.co.uk>

Software

Cool Edit (1995). Syntrillium Software Corporation. Phoenix, Arizona
<http://www.cooledit.com>

Adobe Photoshop (1999). Adobe Systems Incorporated.
<http://www.adobe.com>

QuickTime (2001). Apple Computer, Inc.
<http://www.quicktime.com>