Chapter 8 Testing and Evaluation

8.1 Introduction
This chapter describes the testing and evaluation of the template and the language learning program developed. Section 8.2 reviews the testing that took place at three levels. The concept testing asks the question as to whether the concept of a template for a CALL system for ELs works. Software testing refers to the testing of the produced courseware. It looks at questions such as: whether the web pages are generated correctly, whether all the links work, whether all the audio files exist. The courseware testing reports on the testing of the course itself, the proofing of the languages used in the course (both Nawat and Spanish) and the revision of the explanations used in the course. Section 8.3 covers the evaluation phase of the system. It reviews the objectives and requirements of the system. It reports on the template implementation and whether it works as a language learning program. The system is then evaluated from a software point of view. The results of the evaluation module defined by Hubbard (1996) are presented. This includes the operational description component, the learner fit component and the teacher fit components. Section 8.4 reviews the implementation of the courseware. Section 8.5 highlights some unexpected outcomes of the project. Section 8.6 provides a summary of the chapter.

8.2 Testing
This section covers the testing of the template. It deals with concept testing, software testing and the testing of the courseware itself. Figure 8.1 shows the chronology to the testing phase of the project. CALL testing, as with any testing, is an iterative process. Bugs in the software and errors in the courseware (text, audio and image data) are detected and corrected and the modified courseware is retested. The combination of the waterfall and evolutionary prototyping models adopted in the development of the project (see section 7.2, p116) meant that the development and testing occurred in parallel and were closely intertwined.

![Figure 8.1 Chronology of the test phase of the project](image-url)
8.2.1 Concept Testing

At the highest level, the project had to be tested at the "proof-of-concept" point of view. That is, does the concept of creating a template for developing a CALL program for ELs actually work? The template developed was tested against its original specification: given the EL (Nawat) language specific components in the form of XML data files, audio files with the recorded conversations and the image files with culturally specific images, can it create language learning courseware for the EL (Nawat)? Using these source files, the template software was able to convert them fully automatically into the complete set of web pages that make up the language learning courseware.

The fact that the lessons were prepared and conversations were recorded in advance of the template software development (and were not structurally changed) indicates that the template functioned for the Nawat language learning program.

Another test is whether the template works for other languages. To check that it did, language lessons were developed for Irish (Gaelic) and Akan (a Ghanaian language). This demonstrated that the template was not language specific.

8.2.2 Software Testing

Software testing involves both verification and validation (Somerville, 1996). Verification involves checking that the program conforms to its specification, while validation involves checking that the program as implemented meets the expectations of the user. Static checking techniques include program inspections and analysis. Dynamic techniques (tests) involve exercising the system. The testing process normally has five stages. Firstly, individual units are tested in unit testing. Module testing tests modules (usually a collection of dependent units). Sub-system testing tests collections of modules and often exposes sub-system interface mismatches. System testing tests the system as a whole and finally, there is (user) acceptance testing.

In this project, unit testing was carried out on each individual XSL file, as well as driver and generate file types. Module testing was carried out on each logical unit (e.g. section or explanation module). Sub-system testing mainly involved testing the lesson, dictionary and text information sub-systems. System testing was then carried out for the complete system. Provisional user acceptance testing was done in Ireland and final user acceptance testing was carried out in El Salvador (see section 8.3.5).

When carrying out software testing, there are several strategies that can be adopted. The two most basic strategies are top-down and bottom-up testing. Top-down testing starts with the most abstract component and works downwards while bottom-up testing starts with the fundamental components and works upwards. Top-down testing helps to detect unnoticed design errors at an early stage and can provide a psychological boost to the development team as a limited, working system is available at an early stage in development. Bottom-up testing of the low level system components is almost always necessary.
Both strategies were used in the development of the project. Top-down testing was used to test the overall look-and-feel and structure of the system. It checked that the system hung together and was not missing any key components. Bottom-up testing was used to test each component and module.

As it is more important to test commonly used parts of the system rather than rarely exercised parts, the lesson part of the system was heavily tested. All the links in various orders and combinations were tested. The dictionary was also tested, with complete testing of all the links and then incremental testing when modifications were made.

Testing was made somewhat easier by the use of standard naming conventions and the fact that the web pages were automatically generated. This meant that as long as the conventions were adhered to, testing of one generated instance (e.g. a lesson) could (almost) suffice for testing of all instances, as each instance would be generated in the same way. This meant that once problems were resolved for one instance, they were resolved for all.

8.2.3 Courseware Testing

Overall System Testing

Courseware testing took place on several levels. Firstly, there was the overall system testing. The courseware was tested to ensure that all the conversations were correctly transcribed and that the sound files were correctly matched. All the explanation and vocabulary files were revised for correct and missing content. All the activities and exercises were tested to ensure that the answers were correct and that no incorrect answers sneaked through. The testing of the other system tools (e.g. alphabet, dictionary, and courseware tutorial) were carried out in a similar manner.

Nawat Language Testing

Secondly, the testing of the linguistic content in Nawat was carried out. Although a native speaker had checked the lessons before being recorded, it was still imperative to check the Nawat content, along with the explanations and vocabulary. This testing was carried out by Genaro Ramírez (see section 5.6, p90) who is a native speaker and Dr Jorge Lemus (Universidad de Don Bosco, Universidad de El Salvador), a linguist who speaks Nawat. The testing was carried out separately but similar results were produced. I worked with Genaro Ramírez, checking the correctness of the Nawat and the accuracy of the explanations. Dr Lemus kindly reviewed the contents independently. There were one or two corrections made but fortunately only minimal changes were required.

Spanish Language Testing

Thirdly, there was the testing of the Spanish language used throughout the course. As I am not a native Spanish speaker, this was essential. Several ex-colleagues in the Universidad de El Salvador in San Vicente helped me in this task, especially Lic. Miguel Ortiz. I was initially apprehensive, especially with regard to the explanation parts as I wanted the language used to be easy to understand and informal. Although there is always room for improvement, hopefully most of the errors have been corrected as a result of this phase of testing.
Document Testing
As the printed version of the courseware was generated from the same source files as the online version, errors eliminated in the above testing phases (overall system testing, Nawat language testing and Spanish language testing) were also removed from the printed version.

The User Manual was tested was pilot tested by a non-expert computer user in Ireland. He was able to create a section of a lesson (with limited assistance) using the User Manual (see Appendix K, p261 for the User Manual).

8.3 Evaluation
This section covers the evaluation phase of the project. It reviews the objectives and requirements of the project and evaluates the implementation of the template. It answers the question as to whether the system works. Furthermore, it discusses the software and user evaluation of the project.

<table>
<thead>
<tr>
<th>Evaluation Template</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>different languages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>language learning program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluator 1 (Ireland)</td>
</tr>
<tr>
<td>informal evaluators (Ireland)</td>
</tr>
<tr>
<td>CALL knowledgeable student (Ireland)</td>
</tr>
<tr>
<td>evaluators (El Salvador)</td>
</tr>
</tbody>
</table>

Figure 8.2 Chronology of the evaluation phase of the project

8.3.1 Review of Objectives and Requirements
A brief review is provided here of the objectives and requirements of the project. Chapter 1 contains more complete coverage of the project objectives, requirements and constraints.

Objectives
The main objective of the project was to create a template to enable the development of a CALL program for ELs. In addition, there were secondary objectives. The first was to put Nawat on the computer, to show the people of El Salvador that the language could be part of the modern world and that it was not just a dead, "useless" language. The second aim was Nawat language documentation. The development of a language learning program such as this means that part of the language can be maintained and documented for future reference. The recorded speech is particularly valuable as most of the language resources currently available are in the printed format.
**Requirements**

There were several requirements that had to be borne in mind. The template would have to be modifiable, so that it was easy to correct, add and remove information. It would have to be easy to maintain. It would have to produce language learning courseware that would be easy to use (especially for non-experts) and would encourage people to use it. Another requirement, not usually required in the field of CALL, was the ability to provide the courseware on CD (for those with computers but no Internet access) and in printed format (for those currently without computer access). The template would also have to maximise the use of the courseware (text, audio, and images) available to it.

**8.3.2 Template Implementation**

Obviously, the template had to be checked that it did what it was supposed to do. It was used to create a CALL program for the Nawat language. The requirements of the Nawat language program helped in the development of the template (as it demonstrated the real needs of a language learning program). While Nawat itself has very few accented and non-standard characters, Spanish has several. Thus the ability of the template to handle non-ASCII characters has been tested.

Templates already exist for the development of CALL programs. The aim of this project was to develop a template for ELs. Whether it has achieved this depends on two considerations. Firstly, can the template be used to develop a CALL program for Endangered Languages? Secondly, does it make the creation of such a program possible given the problems faced by ELs? The answer to both of these questions is yes.

Firstly, the production of CALL courseware for Nawat proved that the template could be used to develop a CALL program for Endangered Languages. While Nawat may have something in its favour that not all ELs may have (i.e. some linguistic descriptions, dictionary and alphabet), it is not untypical in its limited number of speakers and their age profile, level of education and social position. Despite these difficulties, it has been possible to create an *ab initio* course for the language.

The second issue to consider is whether it could be used by someone non-technical, who is interested in creating a CALL program for a given EL. It is not so easy to answer this question. While I would like to think this is the case, I cannot definitely claim it to be true. However, when I showed the system to various non-technical people in El Salvador (mainly linguistics or language specialists), they were able to understand the XML file structure and how to generate the HTML pages (see section 7.3, p123 for information on the structure of the system). I accept that these were formally educated people, but I do believe that in most situations the help of university members can provide a bridge to the technology (and also to the computers) and that their help can be very useful in the development of EL resources. They may also have access to the sound and image technology required for the CALL program. Note that while these aspects of the system are desirable, they are not indispensable.

An obvious question to be answered is why an authoring tool was not used throughout the development of the EL courseware. Authoring tools have been successfully used in the production of CALL materials. However, especially in the case of CALL for ELs, they have several disadvantages. While freeware or shareware Authoring Tools do exist, many are commercial products, with commercial prices.
Furthermore, they are often “rich”, substantial systems and may place considerable hardware requirements with regard to main memory, clock speed and graphics which cannot be assumed to be available in the EL setting. Authoring Tools tend to be general-purpose tools (not specifically aimed at the production of CALL materials) and users must learn to master the tool first, which can distract from courseware development.

In contrast, the template developed is lean and free. It is tailored to a specific task – the production of CALL materials. It embodies sound pedagogical principles. It enables rapid prototyping, and because of the clear separation between data and its processing/presentation; it allows CALL developers to concentrate on content rather than the mechanics of production.

Obviously, the template cannot claim to be of the same standard and completeness as a commercial Authoring Tool. However, the template offers advantages especially in the context of ELs. The economic reality of many EL communities means that commercial Authoring Tools are not an option, while the learning curve often associated with some Authoring Tools would not be suitable. The template has a simple developer interface. It is lean in terms of system requirements, which is important in the EL context where the available computers may not be as powerful as machines with higher specifications available in other CALL deployment environments.

The rapid prototyping is useful when developing CALL materials for ELs as previous exposure to and knowledge of CALL may be very limited. While Authoring Tools can provide more pedagogical freedom, it is helpful to provide guidance in the EL context (especially considering that currently available CALL materials for ELs tend not to incorporate generally accepted CALL guidelines – see section 3.6, p48). The template design allows the CALL developer to respond quickly and flexibly to substantial design changes triggered by user feedback (for example, see section 8.3.4). This is particularly important for ELs where local conditions and requirements may vary from “standard” implementations.

The reusability aspect of the template was also demonstrated. The three levels of reusability that were envisaged at the start of the project became a reality. The processing engine was used in the production of the courseware lessons, the linguistic structure was used to produce courseware for different languages (Akan and Irish) and the courseware was reused both in the courseware (e.g. audio portions were used in both lessons and exercises) and on different media.

In keeping with the objective of avoiding "reinventing the wheel", the template used the Hot Potatoes software (Hot Potatoes, 2001) for the creation of CALL interactive exercises. Hot Potatoes is an Authoring Tool specifically for the creation of interactive exercises. Part of the Hot Potatoes software was reverse engineered so that the exercises could be automatically generated from XML data files, rather than having to create each exercises individually online, which is the case with the Hot Potatoes software (see section 7.3.8, p132 for details). Thus, the template managed to integrate existing software and adapt it successfully to its needs.
The anticipated constraints (no funds, and time issues) prevailed. However, the project was completed without any major problems – partly due to foreseeing potential difficulties in advance and being able to plan around them. I believe that my field trips were especially important in this context and I had to deal with real world problems, rather than just trying to image what potential problems there might be. The fact that the courseware was successfully developed given these constraints (which are probably very typical in the EL context) reflects well on the template.

8.3.3 Does it Work as a Language Learning Program?
The aim of the developed CALL program is to enable someone to learn Nawat. I cannot point to someone and say that they have learnt Nawat using this program. However, the feedback I received in El Salvador was very positive (much more so than I was expecting). Staff of the Universidad de Don Bosco and staff and students of the Universidad de E L Salvador were intrigued by the idea of learning Nawat via the computer. I demonstrated the Nawat courseware in the Universidad de Don Bosco and the Universidad de El Salvador (both in San Salvador and San Vicente) in July 2001. The courseware was positively received, both by those with prior experience and knowledge of Nawat and the Pipil community and those without such knowledge. I feel that the audio aspect of the system made a big difference, as they were especially impressed with this part of the courseware. Although there were very few elements in each activity, people responded well to the fun aspect and light hearted approach. People came away from the first lesson saying "yehyek tunal" (hello), and asking me for a copy of the software.

I am very well aware that this initial show of enthusiasm, while genuine, does not automatically translate into real users and learners. However, it far exceeded my expectations and perhaps it may be that start of something positive for the Nawat language. Since my return from my second field trip, I have been informed that people have been making copies of the Nawat courseware. Even if they do not actually learn Nawat, their interest in the language is welcome. It also aids the dissemination of Nawat documentation and raises the social profile of the language.

8.3.4 Software Evaluation

Chosen Technology
A decision was taken early on in the project to use a web style interface. Apart from it being an accepted format, it lends itself to the integration of sounds and images. Rather than use HTML files directly, it was decided that the use of XML technologies would be more appropriate. XML technologies supported the many changes that were made to the screen design during the project development.

During my second field trip to El Salvador, it was suggested that it would be useful to have a grammar/revision section after every four lessons. It was a valid suggestion and one that I had toyed with previously. However, it meant having to write the XML data files (and decide what to put into them), develop the XSL file and put the relevant links into 600 plus web pages.

As the syllabus was originally developed with learning themes and grammar items in mind, that material was available as part of this process. The revision XSL file would have to follow the same pattern as the other pages. The revision file was developed and incorporated in to the relevant XSL file. To add the
appropriate links, the sidebar file was changed and each of the HTML files for the twelve lessons were
generated using the ‘createLesson’ batch file. This could have been a tedious and error prone process, but
fortunately this was not the case thanks to the flexibility of the XML technology and template design.

I returned to El Salvador once the program had been developed. I worked with the Nawat and Spanish
speakers to correct the errors in both languages. Due to the separation of data from presentation, I was
able to apply the necessary changes to the XML files alone, without having to touch the XSL files. It is
an important feature of the template that it is easy to make changes.

Even working with a non-endangered language, this would be a requirement, but with an endangered one
it is even more so. Spelling errors inevitably exist, problems determining whether something is one word
or two, as well as confusion as to the exact meaning of a given phrase in some situations. If it were
difficult to change the text, it might lead to people avoiding making necessary corrections, which would
lead to undesirable consequences.

8.3.5 Courseware Evaluation
The Hubbard (1996) model (see chapter 6, p93) identifies three elements within the evaluation module:
the operational description component, the learner fit component and the teacher fit component. This
section summarises courseware evaluations that took place both in Ireland and El Salvador.

8.3.5.1 Operational Description Component
A system evaluated by the developer alone could not possibly claim that it works. It was not feasible to
do user testing on a large scale, but it was imperative to do some user testing. Thus qualitative (rather than
quantitative) user evaluation was carried out.

There are two groups of users to consider in the operation description component. The first group is
composed of the members of the EL (in this case the Pipil) community. The second group is made up of
Salvadorians who may be one or more generations removed from the EL (Pipil) community.

The needs of the two groups in terms of language learning are similar except that the level of education of
the second group (Salvadorians removed from the Pipil community) would in general be higher and
furthermore, they would have better access to computers and the Internet. The user evaluation carried
out during the project considered these as one group in terms of language learners and two groups in
terms of media requirements for the courseware.

Given the practicalities of the situation (namely that the template development took place in Ireland and
the target users were in El Salvador), Irish people were used to evaluate the courseware in Ireland in order
to facilitate the development process. Their role was to review the courseware from a language learner’s
point of view, to check the screen layout, control options and materials.

Once the initial shell had been developed, a user carried out an evaluation of Lesson 1 (in Ireland). After
a brief explanation of how the system worked, the user was allowed to work through the system at his
own pace. The areas of difficulty were noted and the overall evaluation was positive. One particularly useful observation was with respect to the lesson exercise. While the user enjoyed the activities (which just involved clicking with the mouse), the leap between simply selecting an option and actually producing target language in the exercise was deemed to be too great.

While all four skills (reading, writing, listening and speaking) are important in the language learning process, given current technologies, it is difficult to incorporate user speech production. The activities tested the passive skills (listening and reading) but it was considered important to include language production in the form of writing. Thus an audio element was added to each exercise with the correct answer. The user could use this to help him/her produce the desired response (language production via writing).

Subsequently, there were several informal evaluations (non-CALL colleagues and fellow postgraduate students in Ireland) with the aim of determining whether things were more or less on the right track. Generally the feedback was positive, with the audio part and the interactive activities proving popular.

Once the program had been fully developed, it was evaluated by a final year computational linguistics student (in Ireland) with knowledge of CALL. She provided detailed descriptions of the problem areas with suggested fixes. Nearly all of the items raised were valid (some were already on the "to do" list). Two particularly important observations were made. One was that there were too many icons in each section (there was a link for each section and activity in the lesson, as well as the explanation, vocabulary and help links). The second was that there was some confusion as to whether the explanation covered just that section or all the sections.

As a response, a simpler, cleaner interface was developed for each section, with links just for that particular section being shown. This could have been a major undertaking, but the flexibility offered by the template meant that with some thought and a bit of reorganisation, it was carried out without too much difficulty.

User interface testing in El Salvador (see Appendix E, p187 for evaluation sheet) did not reveal any major design flaws. Informal general user testing in El Salvador provided positive feedback and the users were enthusiastic about the system. As one of the aims of the project was to encourage people to learn Nawat in an interesting manner, it was gratifying that the system was so positively received.

8.3.5.2 Learner Fit Component

When evaluating learner fit, items such as the learners’ preferred style of learning, program and learner focus, language difficulty and feedback must be considered. The evaluation of learner fit was carried out independently by the EL community leader (Genaro Ramírez) and Dr Lemus (Universidad de Don Bosco, Universidad de El Salvador). They checked that the level of language difficulty was appropriate for the target users. They also reviewed the explanations used in the courseware and suggested slight modifications that were then applied to the courseware. With regard to the contents, they particularly
liked the parts that covered areas not previously covered by Nawat materials (e.g. things to do with times).

8.3.5.3 Teacher Fit Component

Although the courseware was principally designed for the independent learner situation, it is important to evaluate the courseware from the point of view of teacher fit. In the EL context, this takes on extra importance as teachers can have a multiplier effect on the courseware. They can add their own data (such as cultural information), add extra lessons and add words to the dictionary. The courseware only covers beginners’ learning items – it is clear more lessons are needed for more advanced courses. It is important that teacher fit is right so that teachers will feel comfortable using the template.

The potential teachers came from the university sector (Universidad de Don Bosco and Universidad de El Salvador) and Concultura (part of the Ministry of Education and Culture). They were positive in their feedback of the system and said that it was an important step forward for Nawat. They liked the web-based interface and the ability to move around the system (via links) at will. Their principal need was to be able to use the template to add data, notably cultural data.

At a workshop on how to add cultural information to the system (Universidad de El Salvador, 29 July 2001), lecturers were shown how to add information to the system. They were not technical people and once they were shown some practical examples, they were able to practise adding data themselves. They expressed satisfaction with the system and were inspired by the potential it offered. Lecturers in both universities (Universidad de Don Bosco and Universidad de El Salvador) will be able to act as content providers and thus, the template will achieve a multiplier effect.

One person (a lecturer in the Literature Department, Universidad de El Salvador) asked whether a feature to check the students’ results could be added to the system. I explained to him that the aim was to encourage students to study not to scare them away, and there is nothing like the word "exam" for causing stress in learners. I told him that the activities were to be self-evaluated, rather than exams. The learners can repeat them as often as they liked (learner autonomy) and as the questions and their answers were displayed in random order each time, simply remembering the correct answer for each question would be not be enough.

8.3.5.4 Comment

It was not possible to carry out user evaluation of the online system with the EL community members in El Salvador. There is no computer available in the community and various efforts to go to the community with a laptop or to bring community members to the nearest town did not materialise. However, the multiple modalities of the developed courseware meant that the printed version was accessible to the EL (and wider) community.

While it might be argued that this indicates the futility of developing such a system if the EL community members cannot use the system, I would make the counter observation that hopefully this will not always be the case and that someday in the not too distant future, they will have access to a computer. Also,
other Salvadorians (some more recently removed from their Pipil ancestry than others) do have access to computers and they can use the system.

A copy of the CD and printed version of the courseware have been made available in the cultural centre in Santo Domingo de Guzmán, where it will be accessible to community members. Both my Nawat informants (Genaro Ramírez and Paula López – see section 5.6, p90) were aware of the potential of not only the language learning aspect of the project, but also the language documentation aspect. They never suggested that it was a bad idea and that the lack of computers in the community meant that developing a CALL program for Nawat was futile. They are people with vision and care for their language and while they may have a very limited understanding of computers, they understand that it can benefit the documentation and teaching of their language.

It is important to note the second group of potential learners – the Salvadorians who are one or more generations removed from the Pipil community. Often they have access to a computer, if not the Internet. They can take a copy of the CD and have access to the courseware by that means. Thus they can gain the benefits of CALL without having Internet access. The Internet access is useful for those without access to their own computer but may have access to the Internet (either in a university or Internet café). A copy of the Nawat courseware is available on my web page (Ward, 2001) and a copy is available on the web page of the Universidad de Don Bosco (UDB, 2001).

8.3.6 Other Objectives

There were two other objectives to this project. One was to show that Nawat could be put on the computer - it could be online, on the Internet, part of the digital age - to show that it was not just a language of the old people. While the mere fact of putting Nawat on the computer will not automatically raise its esteem, it might make a contribution to the process. The objective has been achieved.

Another objective was with an eye to the future. It may be the case that the program may not be used at present due to a variety of factors. However, it can serve as a tool for use in the future. Making this Nawat resource digitally available means that it can potentially serve as a form of language documentation for future generations. An electronic version of the courseware is now available on the Internet and on CD. A printed version also exists. These have been distributed to interested parties in El Salvador and will be hopefully be available to anyone who wishes to access them.

8.4 CALL Implementation

There are three CALL implementations (in the Hubbard sense (1996), i.e. deployment) of the courseware. The first and most important is with the EL community in Santo Domingo de Guzmán. The community can use the printed version of the courseware until they have access to a computer. It may also be possible to bring interested members of the community to the nearest big city, Sonsonate, to access the software in a local secondary school that has computer facilities.

The second implementation rests with the universities in El Salvador. Copies of the courseware are available in the Universidad de Don Bosco and the Universidad de El Salvador. Staff and students will be
able to use the courseware to study Nawat. The third group of users is composed of Salvadorians one or more generations removed from the Pipil community. They can study Nawat in an independent learner mode. Since my return from the second field trip to El Salvador, this group has expressed great interest in the courseware, with friends being asked for copies of the CD.

The courseware is available on three different media (Internet, CD and print) to make it more accessible to a wider audience. While a digital medium is sufficient in the usual CALL context, a printed version was essential in the EL one.

8.5 Unexpected Outcome

When I went to El Salvador for the second field trip (July 2001), I hoped that a few people would be mildly interested in the program. I was quite surprised by the reaction. Nearly all the people who saw it were impressed. Dr Lemus organised a seminar on Nawat at the Universidad de Don Bosco (20 July 2001). I made two presentations: one was a general introduction to the Nawat courseware and the other was a presentation on the use of XML technologies in the development of web pages. The reaction was very positive. The director of the cultural section of the Ministry of Education and Culture (Concultura) was impressed and said that it was an important step for the Nawat language. I was invited to formally present the courseware to the president of Concultura at the headquarters of Concultura in the Ministry of Education and Culture, which I willingly did on 30 July 2001. The minister of culture said that the courseware could act as a catalyst in the search for a standard alphabet for the Nawat language and thanked me for my contribution to the documentation and preservation of the Nawat language. I also received several emails about the Nawat courseware.

During the seminar in the Universidad de Don Bosco, I was interviewed for national television (TV12). It was not something that I particularly relished but it was done in the spirit of promoting the Nawat language. The questions centred around the history behind the project and the motivation for developing Nawat courseware. It was a positive interview, with none of the negative stereotyping that sometimes plagues ELs.

After the seminar, there was an article in the national paper (El Diario de Hoy) about the Nawat courseware (23 July 2001). A copy of the page, together with an English translation is available in Appendix G, p190. The headline reads “Nawat on the Internet” and briefly outlines the history of the project and what the courseware contains. The article is accompanied by a section which highlights the current plight of Nawat.

I was interviewed by the newspaper of the Universidad de El Salvador, in which I outlined the contents of the Nawat courseware. CALL is not a major element of language learning in El Salvador and the combination of Nawat and CALL impressed most people who saw the system. I also did a radio interview for the university radio station (30 July 2001), which lasted 15 minutes and covered many topics – some specifically about the course and some about Nawat in general. The presenter ended the program by teaching the listeners yehyek tunal (hello).
Lecturers in the Literature Department in the Universidad de El Salvador were interested in adding cultural information. They have undertaken research on the Pipil culture and wanted to incorporate this into the courseware. At present, the information is only available in printed form. I organised a workshop to show them how to add information to the culture section of the Nawat courseware (27 July 2001). They intend to add their information to the system.

I believe that actions speak louder than words, so only time will tell if the ideas for expanding the pages, involving students from the universities to help gather material (amongst others), will come to fruition. At least, some sort of process has been set in motion. It may be some time before the idea becomes a reality but it has provided a potential forum for disseminating the already collected cultural information.

The project has aroused a certain level of awareness of the language and has provided a public relations opportunity for Nawat. The TV, radio and press coverage provided positive publicity for Nawat rather than just lamenting its imminent demise (which tends to be the focus of other Nawat articles). The universities (Universidad de Don Bosco and Universidad de El Salvador) are interested in adding to the courseware. As was mentioned to me several times during my second field trip, what this means is that the seed has been sown and much work remains to be done. However, as Confucius is alleged to have said “A journey of a thousand miles, begins with a single step”; the Salvadorians have been provided with the courseware, but now it is up to them to decide how it develops.

8.6 Summary
This chapter covered the testing and evaluation phases of the project. It discussed the testing of the project from several different points of view. The "proof-of-concept" testing was the most fundamental testing that was necessary for the project. As with any software development, the program had to be tested from a software point of view. However, even if the concept works and the software is tested as required, the developed courseware is virtually useless if its contents are not correct. Thus, the courseware testing was an important aspect of the testing phase.

Evaluation has often been a neglected part of the CALL development process. However, this project has endeavoured to incorporate it at various stages during development, from early user evaluation of a system prototype to user evaluation in the intended implementation situation (in this case, El Salvador). The various aspects of evaluation that are important in a project such as this (template, software and user evaluation) were all described in this chapter. Testing and evaluation with members of the EL community still needs to be done with the online version but that printed version has been distributed to the community and the feedback has been positive.