ReCALL **14** (2): 285–294. © 2002 Cambridge University Press
DOI: 10.1017/S0958344002000629 Printed in the United Kingdom

Reusable XML technologies and the development of language learning materials

MONICA WARD
School of Computer Applications, Dublin City University, Dublin 9, Ireland
(*email: Mward@compapp.dcu.ie*)

Abstract

There are many CALL resources available today but often they need to be adapted for level or culture. CALL practitioners would like to reuse currently existing material rather than reinvent the wheel, but often this is not possible. Thus, they end up building CALL material, both language content and software, from scratch. This is inefficient in terms of CALL practitioners' time and as Felix (1999) points out, there is no point 'doing badly' what has already been done well. Why can't we reuse what already exists? Often the language content is hard-wired in the software and cannot be modified or the CALL material comes as an executable which is hard or impossible to change. Authoring tools can provide a degree of flexibility, but often focus on particular parts of the language learning process (e.g. interactive exercises) and the associated language content cannot, in general, easily be exported into other formats and presentation software. One solution to this problem is provided by XML technologies. They provide a strict separation between data and processing. Thus, three types of reuse are possible: reuse of the data processing engine (i.e. the XSL processing files), reuse of the language content structure (i.e. the XML data files) and reuse the linguistic resources (i.e. the language content). In this paper, an example is given of a CALL template that has been developed using XML technologies. The template provides a structure into which the language content can be slotted and a processing engine to act upon the data to create CALL material. The template was developed for the production of CALL materials for Endangered Languages (ELs), but could be used for MCTLs (Most Commonly Taught Languages) and LCTLs (Less Commonly Taught Languages) also. It has been used to develop a language learning course for Nawat (a language of El Salvador), Akan (a Ghanaian language) and Irish, demonstrating the reusability of the language content structure as well as proving the reusability of the processing engine provided by the template. A further reusable feature is the ability to create courseware in different media (Internet, CD and print) from the same source language content.

1 Introduction

The need to avoid reinventing the wheel is something CALL practitioners are very much aware of. Whether the language being taught is amongst the world's MCTLs, LCTLs or ELs, CALL practitioners would like to build upon what already exists without having to

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

start from scratch. However, often changes need to be made to the available material to adapt it to local needs or a different target group. This is usually difficult to do because often proprietary software is used to produce CALL programs or the language data has been encoded in some format that is difficult to modify.

With the increasing permeation of the WWW into all aspects of modern life including language learning, there is a movement to put manageable standardised and reusable structures in place for the production and maintenance of web pages. XML (eXtensible Markup Language) and its related technologies are emerging as the vehicle for this new structure. This article outlines XML and the benefits it can bring to the CALL community.

The first section outlines why we should try to avoid reinventing the wheel. It reviews why it is often hard to reuse the material that is currently available. The second section outlines the philosophy behind XML and how it can be used in the development of CALL materials. Section three presents a template developed using XML technologies. The template provides both a structure for the language content and a general processing engine that operates on the language content to produce CALL material.

1.1 Why we need to avoid reinventing the wheel

Felix (1999) suggests that CALL resources should be pooled on a world-wide basis as there is no point 'doing badly' what has already been done well. For the world's MCTLs (English, French, German, Spanish) there are now many resources available in the CALL domain. For LCTLs there is less material available, although this is rapidly changing.

In the case of ELs, the use of CALL programs is, understandably, minimal. Generally, the primary focus is on language documentation while the remaining speakers are still alive. Often, there are only a few people involved in the preservation of an EL. This small group may include community members and a linguist – but rarely a CALL specialist. In her survey on the use of the web for the teaching of ELs in North America (mainly Native American languages), Buszard-Welcher (1999) found that the presentation of the material that is available tends to reflect the teaching practices that the EL elders experienced when they were school children. Many of the findings of SLA research and good CALL practice have yet to be incorporated into the sites. It would be useful for EL communities to avoid retracing the history of CALL development in the past few decades. It would be helpful to have access to a template which supports development of CALL materials that incorporate current good practices from the field of CALL.

1.2 Why people try to reinvent the wheel

Most CALL practitioners would prefer not to have to create CALL materials from scratch. However, often they end up doing precisely this. There are several reasons for this. It may be due to lack of knowledge about the existence of, or lack of access to, already developed material. However, often a more fundamental reason is lack of suitable material. Bangs and Shield (1999) point out that there are a limited number of programs available which are both pedagogically sound *and* adaptable for end user needs and purposes. A further reason is that a program may not be available for the desired learner level or L1/L2 combination.

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

1.3 Why it is hard to reuse what is currently available

Most CALL practitioners prefer to focus on the teaching, learning and linguistic aspects rather than software development. Many practitioners would like to be able to use the good material that is already available. They would like to be able to adapt it for their own local requirements. They may want to change some things that they are not pedagogically comfortable with. They may also want to add extra material.

Although practitioners may wish to modify an existing CALL program, it is often difficult, if not impossible to do so. If the CALL program comes as an executable, there is often no scope to change the contents. Even if the source code is available (together with the linguistic content), it is often hard to modify without specific technical expertise. What is worse, often the language content is inextricably linked with the code and the two cannot be separated.

One approach to this problem is to use an authoring tool. Authoring tools permit the creation of CALL materials, without the need for deep/substantial technical knowledge on the part of the developer (e.g. Hot Potatoes 2001). Indeed, authoring tools have been used to create many CALL programs (Levy, 1997). However, there are several problems with authoring tools. One is that they can have a steep learning curve. Another is that they may have to be purchased or have licensing restrictions (although freeware and shareware programs do exist). They are often aimed at the creation of a substantial system. They may also have demanding system requirements (e.g. main memory, fast clock speeds, fancy graphics etc). In addition, they may require the language data to be encoded in a system specific way that does not support portability (Davies, 1998). This limits the re-use potential of both the data and the processing engine.

2 What is XML

XML (www.w3.org/xml) is a set of guidelines for designing text formats for data that is easy to produce and read, that are unambiguous, extendable and platform independent. XML is a method for putting structured data into a text file. XML looks a little like HTML but, unlike HTML, only defines data, not the format of how that data is to be presented. There is now a family of XML modules that work with XML files. An XML file specifies the structure of data, while XSL (eXtensible Stylesheet Language) (http://www.w3.org/Style/XSL/) processes and formats the XML encoded data. Figure 1 shows an example of how XML files and XSL files are combined to produce HTML pages.

XML allows data to be organised in whatever way one chooses. By contrast, HTML comes with a fixed, pre-defined set of tags (for example, <body> or <p> for paragraph). With XML, it is possible to define tags which can be used to specify the meaning of data. Let us illustrate XML with an example.

Suppose, we wanted to design a mini-dictionary. The entry for a given word would contain the pronunciation, a gloss, the meaning and an example. Each data item is identified by a start and end tag (see Figure 2).

Later in the development of a CALL application, we might decide to add a category (hypernym) element (for example, ‘dog’ can belong to the ‘mammal’ category). To do

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

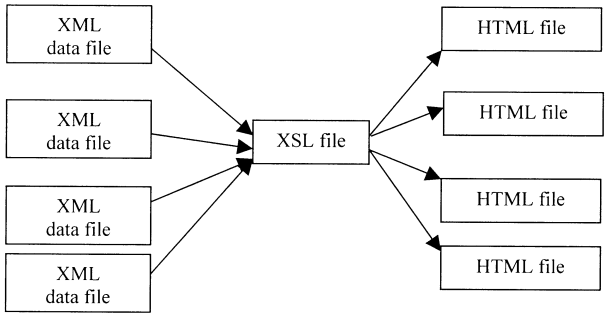


Fig. 1. How XML files and XSL files are combined to produce HTML files.

```
<entry>
        <word>dog</word>
        <pronunciation>dog</pronunciation>
        <meaning> highly variable domestic mammal (Canis familiaris)
        </meaning>
</entry>
```

Fig. 2. Sample XML file for a dictionary.

```
<entry>
        <word>dog</word>
        <pronunciation>dog</pronunciation>
        <meaning> A highly variable domestic mammal (Canis familiaris)
        </meaning>
        <category>mammal</category>
</entry>
```

Fig. 3. Sample XML file for a dictionary with additional tag.

this we could simply add an additional tag to our word entry (see Figure 3).

To display the dictionary information on a web page, we need to write an XSL file to process the information and specify its format. Figure 4 shows some sample XSL code for a dictionary. The tags from the XML data file are highlighted in bold. For further information about XML and its related technologies see http://www.w3.org/Style/XSL.

2.1 The benefits of XML

XML is easy to read and understand and can be processed by computers. It is an open standard, is public domain, not tied to a particular technology and is browser and platform independent. No specialist editor is needed to work with XML files (although there are several XML editors that make it easier to work with XML data). Data stored in XML files can be converted into many different formats. There are free processing tools available (e.g. xt http://jclark.com/xml/xt.html) which use XSL files to convert XML

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

```
<xsl:template match="dict">
      <h2> Basic dictionary</h2>

      <-- for each entry in the dictionary -->
      <-- print the word and its pronunciation -->

      <xsl:for-each select="entry">
      <table>
      <tr>
      <td>
            <xsl:value-of select="word">
      </td>
      <td>
            <xsl:value-of select="pronunciation">
      </td>
      </tr>
      </table>

      </xsl:for-each>
</xsl:template>
```

Fig. 4. Sample XSL code for a dictionary.

files from one format into another (e.g. from XML into HTML).

HTML is not difficult to learn and there are a number of web editors and web development kits available that make the task of developing web pages easier.
However, there comes a stage when a language learning site features a substantial number of pages and the site developer wishes to make a global design change. This could be motivated by order of presentation of information within a page or the page layout (for either pedagogical or aesthetic reasons) or selection of information from a generic source. This may involve changing several tens if not hundreds of individual pages – not a pleasant task. Using XML, the format of the presentation of data needs to be changed only once (in the XSL file) and the HTML pages can be regenerated automatically to produce the desired result.

WELL (Web Enabled Language Learning) is becoming an important part of the CALL world. The reasons for this are many, not least because of the relative ease of producing web pages. XML technologies can be used to produce WELL material.

XML allows structure to be imposed on data and supports the separation of data from processing, thus permitting meta-level usage of the underlying data as well as data processing components. This inherent structure lends itself to the development of clean, modular CALL materials.

Data can be stored in XML files just like in a database, without the need for setting up schemas. XML files contain machine-readable content information that opens up new possibilities for search engines and 'intelligent' interpretation of data (for example, examples of the subjunctive in Spanish could be marked up and subsequently extracted from an XML file). With simple HTML files, it is not possible to extract specific information such as this in a straightforward manner – with XML, this is a possibility.

XML allows the separation of content from presentation (Mills, 1999), or, in more computational terms, the separation of data from processing. XML tags describe the meaning of the data. XSL style sheets describe how the information should be presented. The same data can be presented in several different formats (for example, a full and a summary version, or an on-line and a printed version). Students may like to look at an overview of a topic before studying it in detail – XML allows for this possibility.

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

XML supports Unicode, which caters for most of the world's writing systems[1] . This is especially useful for languages that do not use commonly supported character sets. Multilingual documents are also supported. Thus, a course could be developed to teach Greek to both native English and native French speakers.

XML documents can contain any possible data type. Thus items that are useful in CALL applications, such as multimedia files and active components such as Java applets can be easily stored in XML files. If data is already stored in some other format, for example in a database, it can be easily converted into an XML file. Thus information already available can be reused – you don't have to start all over again.

XML is extensible, there are no fixed tags (as in HTML) and new tags can be created as needed. Domain-specific extensions and vocabularies are ways of exploiting XML. For example, specific extensions are being developed for mathematics (MathML, 1997) and music (SMDL, 2001). There is currently research being undertaken in the development of XML strategies for web-based language documentation and description (LEW, 2000). The main idea is to develop an infrastructure for collecting, storing and disseminating linguistic information. Some of the ideas emerging may prove useful to the CALL community.

A template developed using XML technologies is leaner than an authoring tool (in terms of size and cost). Authoring tools tend to be general purpose tools and the user must learn to master the tool before being able to develop courseware. An XML based template can be tailored to a specific task and embody sound pedagogical principles in the development of CALL materials. It also enables rapid prototyping as it is easy to modify. The language content is concentrated in the XML files and can easily be adapted to specific needs.

2.2 The downside of XML

While XML does provide many benefits, it is not without its downside. XML itself is quite simple to learn but XSL is more complex. XSL determines how the XML data is to be presented. XSL is the 'engine' that can convert the XML data files into HTML files. Initially, the help of XSL experts is required to write XSL files. However, once an XSL file has been written, it can be used to process any XML file that has the required structure.

I believe that XML holds great potential in the development of CALL materials. It has been described as the way of the Internet and given the importance of the Internet and WELL to the future of CALL, it is a development that cannot be ignored.

3 XML and CALL – a working example

The benefits of XML have been explained, but how can we avail of this technology in the development of CALL materials? In this section, a working example is presented to demonstrate the potential of XML for CALL applications.

A framework for developing a CALL program has been developed based on XML technologies. It contains templates for lessons, a dictionary, grammar revision, an alphabet and cultural information. The structure of each lesson is defined in the XML files.

[1] Felix (1998b) provides a short account of Unicode as a mechanism for handling multiple character sets in web based language learning.

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

Each lesson has several sections and each section has several parts, including a conversation part, audio files, an explanation, etc. Elements can be added as needed. If a particular lesson needs to have something different from other lessons, that can be catered for, either by the addition or deletion of information. Due to the separation of data and its presentation, CALL practitioners involved in the preparation of materials can concentrate on the lesson content, rather than the technical issues. The template embodies important CALL principles such as the facilitation of learner autonomy and the multimodal presentation of information to address different learning style preferences.

The framework was developed with Endangered Languages in mind, but obviously it could be used for any language. It was used to develop a CALL program for the Nawat language of El Salvador (Ward, 2001). Figure 5 shows a portion of a conversation developed using the template. It records the name of the speaker (<who>), what s/he is saying (<what>) and what it means in English (<english>). The developed template, with its language content structure and processing component can be reused for any given language. It does not offer all the bells and whistles associated with cutting edge CALL programs, but it does provide a basis for the development of reusable CALL material.

One of the nice features of working with XML is that it does not enforce an 'all-or-nothing' approach. In the project under discussion, materials were developed gradually and added over time. For example, lesson prototypes were built with slimmed down explanations and more complete explanations could be added once the 'shape' and 'feel' of the lesson had been determined.

Another feature of XML technologies that was especially useful during the project was the ability to present the same data in different formats. For example, the dictionary can be viewed in basic or full mode. The user can choose between a brief gloss or a full explanation of a word, along with a sound file. Future enhancements could include words presented according to topic (for example, related words) or syntactic type (for example, verbs). Researchers working in the field of linguistics are considering XML formats for the world's dictionaries (Bell & Bird, 2000). When standardised formats begin to emerge, they can be integrated with and reused in a CALL template.

A very useful feature of the XML technologies, especially in the context of ELs, is the ability to present the same underlying data in different formats. The template enables the production of online CALL materials, a CD for those who may not have internet access

```
<content>
      <p1>
      <who>Genaro</who>
      <what>yehyek tunal.  ken tinemi?</what>
      <english>Good morning.  How are you?</english>
      </p1>
      <p2>
      <who>Paula</who>
      <what>naha ninemi yek.  wan taha?</what>
      <english>I'm fine.  And you?</english>
      </p2>
      <p1>
      <who>Genaro</who>
      <what>nusan yek padiux.</what>
      <english>I'm fine too thanks.</english>
      </p1>
</content>
```

Fig. 5. Sample conversation using the template.

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

and a printed version of the language learning courseware. The ability to generate a printed version is particularly important and imperative for EL communities where there may be no access to a computer.

The thirty-six sections of the Nawat courseware (12 lessons, 3 sections in each lesson) were generated by one base XSL file for section information (see Figure 6). This demonstrates the reusability feature of the processing engine of the template. The reusability of the language content is demonstrated by the reuse of the dictionary information to produce either a basic or a full dictionary entry for a word (see Figure 7). Figure 8 shows an alternative reuse of the language content, producing a HTML page and a printed dictionary page from the same source data.

The third reusability dimension was the creation of language learning lessons in Akan (a Ghanaian language) and Irish (Gaelic). The same processing engine was used and the language content structure for each language was populated with the different required data (see Figure 9). Thus, both the language content structures and the processing engine of the template were demonstrated to be reusable. They avail of the underlying reusability of XML technologies, with its clear separation of data, processing and presentation.

McCarthy (1999) alludes to the problem of CALL materials becoming obsolete

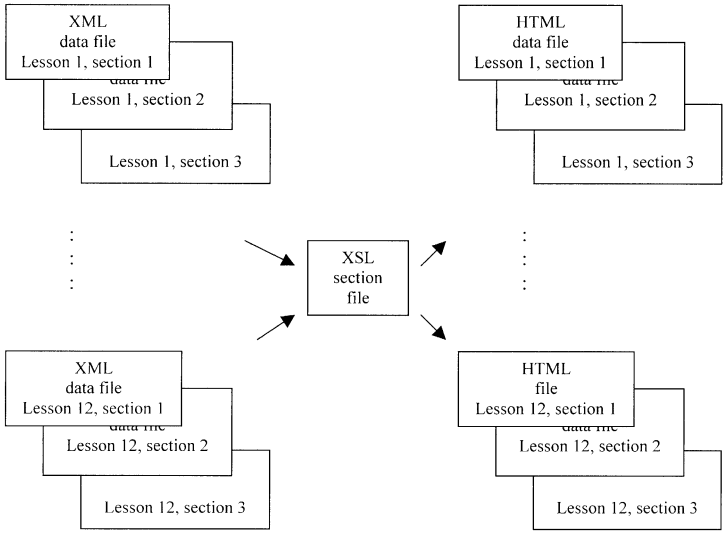


Fig. 6. Reuse of the processing engine to produce the section pages of the courseware.

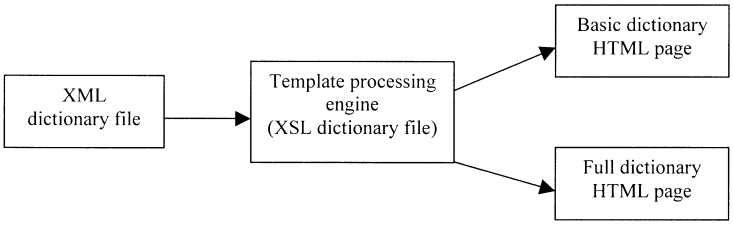


Fig. 7. Reuse of language content to produce different dictionary pages.

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

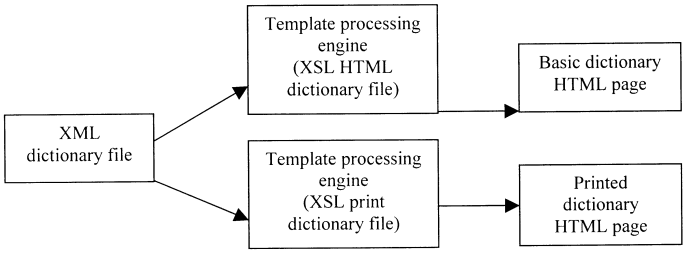


Fig. 8. Reuse of language content to produce dictionary information on different media..

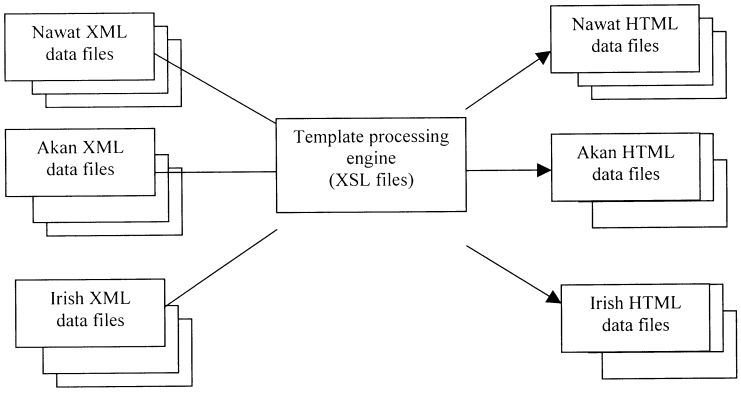


Fig. 9. Reusing the language content structure and processing engine for different languages.

quickly. XML allows for conversion between different data formats, so XML technologies could be used to convert current materials into a new format or incorporated into future materials (this is especially important in the case of Endangered Languages).

4 Conclusion

CALL needs to take into account advancements in its related fields (Maingard, 1999). Levy (1997) refers to the fact that CALL researchers only function within their own domain, seeming to ignore significant developments in other areas. However, given the multidisciplinary nature of CALL, it is difficult to keep abreast of changes in other domains. XML is an emerging technology which could be of benefit to CALL. It provides a simple, solid and clean basis upon which applications can be developed. Rather than reinvent the wheel, XML provides the potential for the creation of reusable templates both for data and processing (be it lessons, dictionaries or whatever). CALL practitioners could have access to tools that incorporate current best-practice instead of having to start from scratch. They could use these templates and adapt them to suit their own specific requirements. XML technologies have something to offer the CALL community regardless of the language being taught. A template for a CALL program has been developed using XML technologies and has successfully been used to create language learning lessons for Nawat (an Endangered Language in El Salvador). Lessons have also been developed in Irish (Gaelic)

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press

and Akan (a Ghanaian language) – demonstrating that an XML technologies-based template can be reused to create CALL materials, without the need to 're-invent the wheel'.

References

Arneil, S. and Holmes, M. (1999) Juggling hot potatoes: decisions and compromises in creating authoring tools for the Web. *ReCALL* **11** (2): 12–19.

Bangs, P. and Shield, L. (1999) Why change authors into programmers? *ReCALL* **11** (1): 19–29.

Bell, J. and Bird, S. (2000) A Preliminary Study of the Structure of Lexicon Entries at the Workshop on W*eb-Based Language Description and Documentation*, 12–15 December 2000, Philadelphia, USA. http://www.ldc.upenn.edu/exploration/expl2000/papers/bell/bell.html (last accessed: 18 September, 2002)

Buszard-Welcher, L. (2001) Can the web help save my language? In: Hinton, L. and Hale, K. (eds.) *The Green Book of Language Revitalization in Practic*e, San Diego: Academic Press, 331–348. http://www.potawatomilang.org/Reference/endlgsweb_update.htm [last accessed: 29 March 2001]

Davies, G. (1998) Four approaches to authoring CALL materials. Paper presented at *EUROCALL 1998*, University of Leuven, Belgium, September 1998. abstract available at: http://www.camsoftpartners.co.uk/author.htm (last accessed: 18 September 2002)

Felix, U. (1998a) Virtual language learning: potential and practice. *ReCALL* **10** (1): 53–58.

Felix, U. (1998b) *Virtual language learning: finding the gems among the pebbles*. Melbourne: Language Australia.

Felix, U. (1999) Exploiting the Web for language teaching: selected approaches. *ReCALL* **11** (1): 30–37.

Garrett, N. (1998) Where do research and practice meet? Developing a discipline. *ReCALL* **10** (1): 7–12.

Levy, M. (1997) *Computer-Assisted Language Learning – Context and Conceptualisation*. Oxford: Clarendon Press.

Lew (2000) *Linguistic Exploration Worksho*p: Web-Based language Documentation and Description http://www.ldc.upenn.edu/exploration/exlp2000 (last accessed: 18 September 2002)

McCarthy, B. (1999) Integration: the sine que non of CALL. *CALL-EJ Online* **1** (2) http://www.clec.ritsumei.ac.jp/english/callejonline/4-2/mccarthy.html (last accessed 18 September 2002).

Maingard, C. (1999) Evolutionary epistemology in language learning: possible implications for CALL. *ReCALL* **11** (1): 80–92.

MathML (1997) Mathematical Markup Language http://www.w3.org/TR/WD-math-970515 (last accessed 18 September 2002).

Mean, J. (1996) Session B: Pedagogy and Materials. Report of the Less Commonly Taught Languages Summit: September 20-21, 1996. *CARLA Working Paper Series* #9, Center for Advanced Research on Language Acquisition, University of Minnesota, Minneapolis.

Mills, D. (1999) XML and Language Learning: What's in Store? 'Lab, Classroom and Beyond: Evolving Technology in Language Education', *IALL* 1999 http://www.iei.uiuc.edu/xml/ (last accessed 18 September, 2002).

Singleton, J., Keane, J. and Nkwenti-Azeh, B. (1998) CALL meets software engineering: towards a multimedia conceptual dictionary. *ReCALL* **10** (2): 33–43.

SMDL, 2001 *Standard Music Description Language.* http://www.oasis-open.org/cover/genapps.html#smdl (last accessed 18 September 2002).

Ward, M. (2001) *The Nawat Language Learning Cours*e. http://www.compapp.dcu.ie/~mward/nawat.html (last accessed 18 September 2002).

https://doi.org/10.1017/S0958344002000629 Published online by Cambridge University Press