Web Access to a Lexical Database using VB/Access CGI Programming

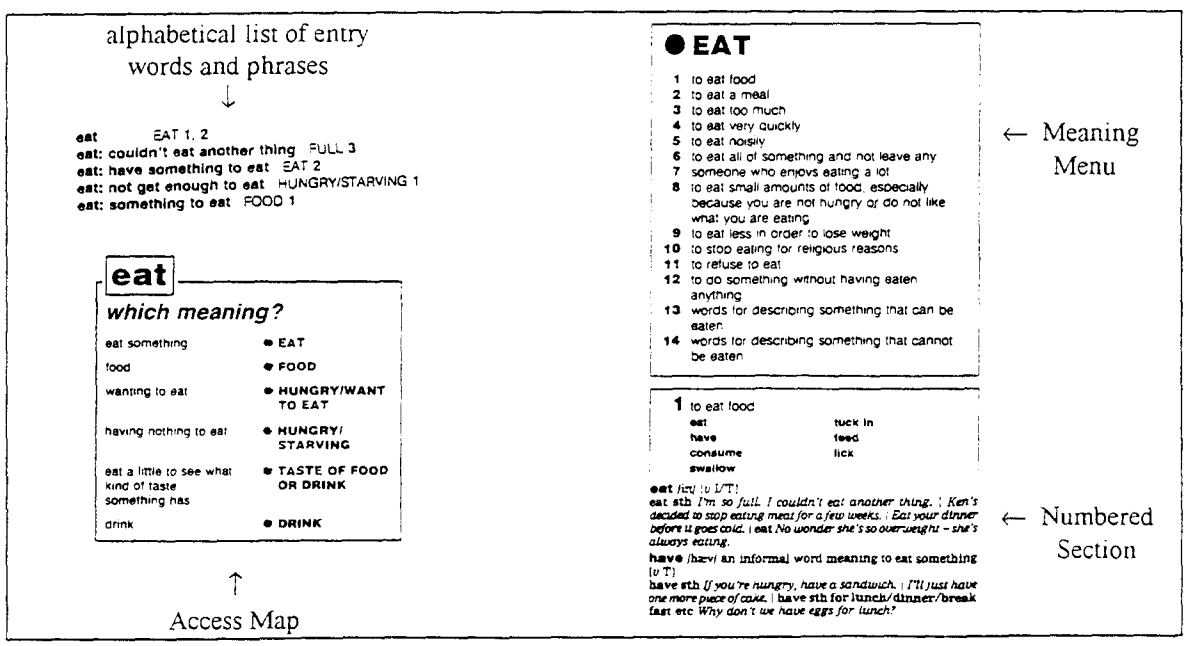
Jonathan J Webster
City University of Hong Kong

Abstract

In this paper I report on the development of an application in which HTML forms serve as a front-end to a lexical database. Lexical information and data retrieval strategies are based on the Longman Language Activator. A Visual Basic CGI application connects a front end HTML form with the back-end relational database implemented using Microsoft Access. Three aspects of the application are discussed in this paper: (1) the lexical database; (2) the HTML front end; and (3) the Visual Basic CGI programming necessary to connect (1) and (2).

The Lexical Database

The source for the lexical database implemented for this application is the Longman Language Activator (LLA), subtitled *The World's First Production Dictionary*. The LLA was selected for its unique organization of dictionary information which is intended to make it easier for the user to find the right word or phrase for a particular context. The LLA is organized on one level according to 'Key Words' or concepts and on another level by the words and phrases which realize these concepts. The 1052 Key Words in the LLA are said to account for the basic concepts from the core of English. For each Key Word, there is what is called a 'Meaning Menu' consisting of numbered sections corresponding to major aspects of the Key Word. The numbered sections follow. Each with another 'menu' of words and phrases. Further detail about each word/phrase — pronunciation, part-of-speech, definition, examples — is provided below this menu of words and phrases. The same words/phrases presented in the numbered sections for a given Key Word are also listed alphabetically throughout the dictionary. One can easily distinguish entry words and phrases from Key Words by the fact that entry words and phrases are in lower-case, whereas Key Words are in upper-case. In addition, the LLA includes what it calls 'Access Maps'. Access Maps identify Key Words related to an entry word.



The LLA adopts three access strategies: first and foremost by concept using the Key Words, second by entry word/phrase, and third by the access maps. The user may start with a Key Word or concept, such as *EAT* and proceed directly to the meaning menu for that Key Word. From the meaning menu, the user identifies the numbered section whose meaning corresponds most closely to their own intended meaning. Next, (s)he looks at the menu of words and phrases in the appropriate numbered section. In order to ascertain which word/phrase best expresses their meaning in a given context, the user may need to look further at the details about those words/phrases in that numbered section. The second strategy begins with an entry word/phrase. Next to each entry word/phrase is a list of Key Words and the numbered section(s) where that entry word/prhase may be found. Also, the list of entry words/phrases contains examples of word sequences (idioms, fixed expressions, habitual collocations) which include that entry word. For example, the entry word **eat** occurs by itself and also in such sequences as **couldn't eat another thing**, **have something to eat**, etc.

eat	EAT	1,2
eat: couldn't eat another thing	FULL	3
eat: have something to eat	EAT	2
eat: not get enough to eat	HUNGRY/STARVING	1
eat: something to eat	FOOD	1

Finally, the user may use an access map to answer the query 'which meaning?'. With the exception of this third access strategy, the other two strategies mentioned above — by either Key Word or entry word/phrase — have both been implemented as part of this experiment in facilitating web access to a relational database, implemented in MS Access, which mirrors the organization of dictionary information in the LLA.

WWW and VB/Access CGI programming

Figure 1 shows the home page for this application from which the user can initiate a search by Key Word or entry word/phrase. To begin a search by dictionary entry, the user may enter a word or phrase in the text box provided, e.g. *eat,* and click on Search for Entry Word.

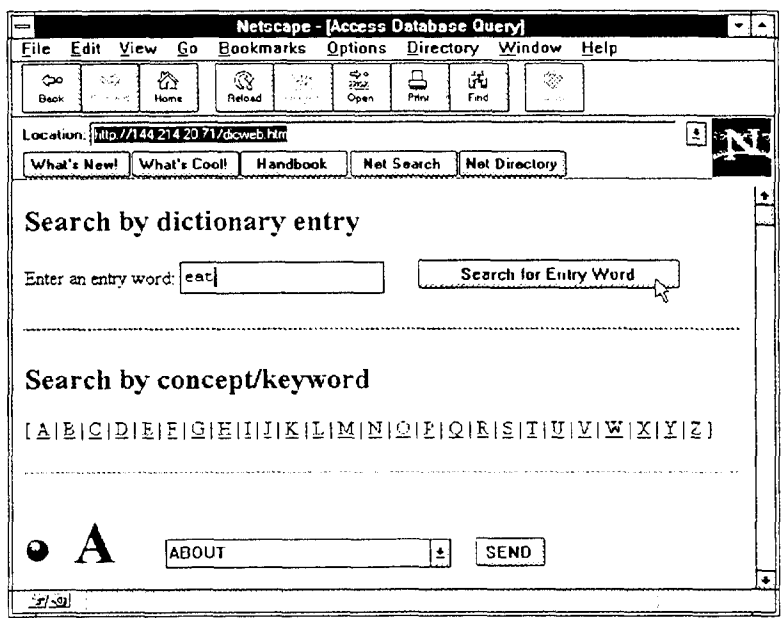


Figure 1: Initiating search by entry word *eat* from home page.

The home page is in fact several forms, the first and top-most consisting of the field into which the user enters a word/phrase to search for and a submit button for transmitting a URL request to the web server. The URL request is a VB CGI exectuable program which queries the Access database, and returns the word list information for the search word. Figure 2 shows the returned web page generated by the VB CGI program. The design of this program is modelled after examples of VB/Access CGI programming provided by Robert Denny, the designer of WinHTTPD and the Windows CGI, and also examples discussed in Heslop and Budnick (1995). Basic CGI initializing operations for this application are handled by Robert Denny's module, CGI.BAS.

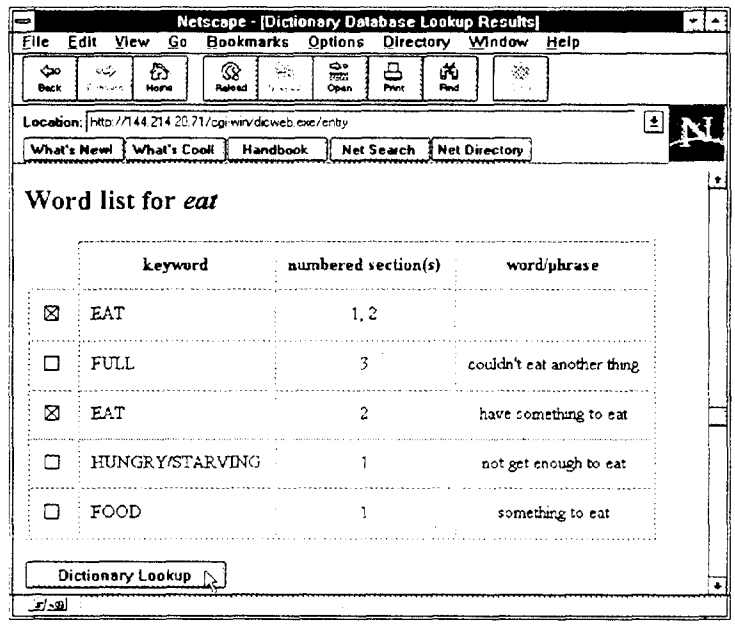


Figure 2: Word List data for *eat*

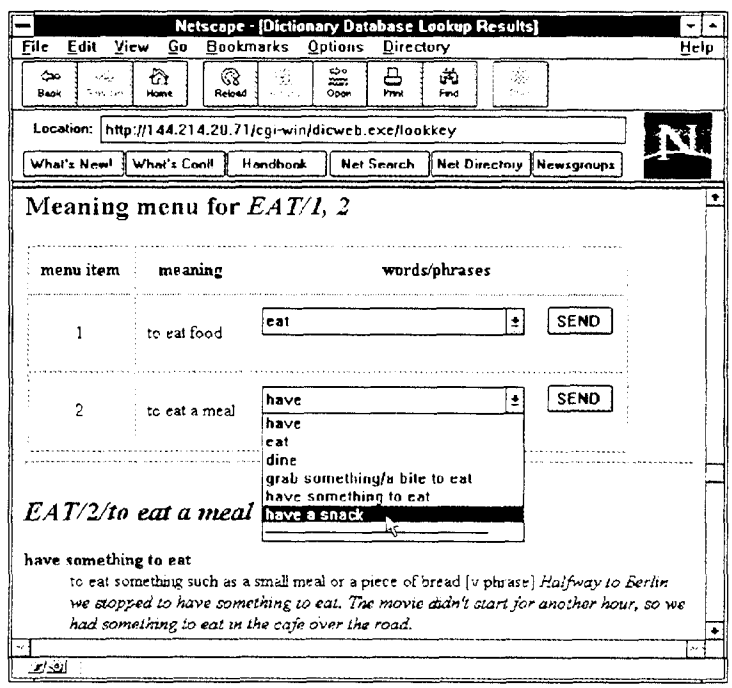


Figure 3: Follow-up to Word List page

In the Word List page (Fig. 2), the user may tick the checkbox for one or more rows in the table to query the database further. For example, a tick in the the first row of the table in the Word List page will generate the Meaning Menu for ***EAT***, including the numbered sections 1 and 2. A tick in the third row of the table on the Word List page generates the dictionary information for the phrase **have something to eat** in numbered section 2 of the Key Word ***EAT*** (see Figure 3). From the Meaning Menu for ***EAT/1,2***, the user may select a word/phrase from the menu of words and phrases, represented as a drop-down list (Fig. 3). Again, a request in the form of any VB executable is sent to the server to query the database and return the dictionary information about the selected word/phrase, e.g. **have a snack**. See the page generated by the VB program in Figure 4.

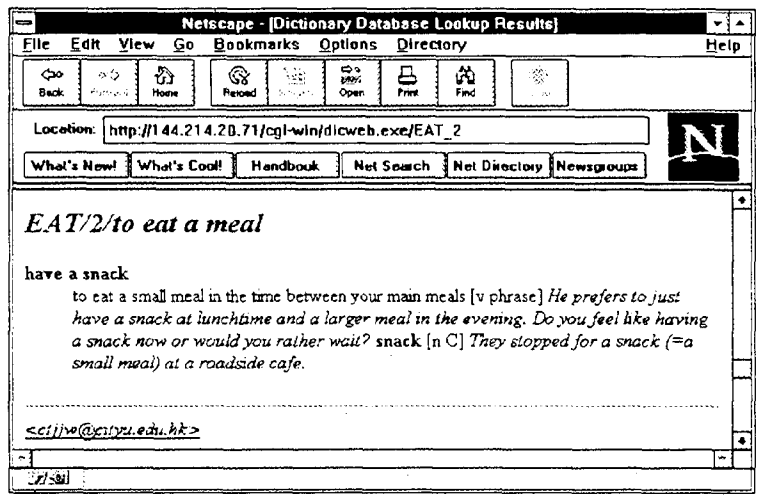


Figure 4: Dictionary data for the phrase, **have a snack**.

Returning to the home page, we can see how the user might conduct a search by Key Word. Figure 5 shows the drop-down list of all the Key Words beginning with the letter "E". Selecting ***EAT*** and clicking on the SEND button will result in a query to the database located on the server, generating a page showing the Meaning Menu for ***EAT*** (see Figure 6). Again, from the Meaning Menu (Fig.6), the user may select a word or phrase from the drop-down menu of words and phrases. Clicking on the SEND button posts the request to the server. The request, the same VB executable as described above, retrieves the details about the word/phrase from the database located on the server, and generates a page to be returned to sender containing the requested data.

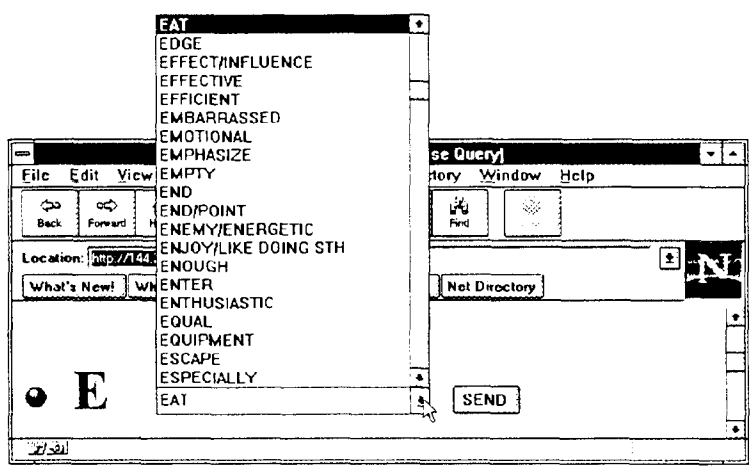


Figure 5: Initiating search by Key Word

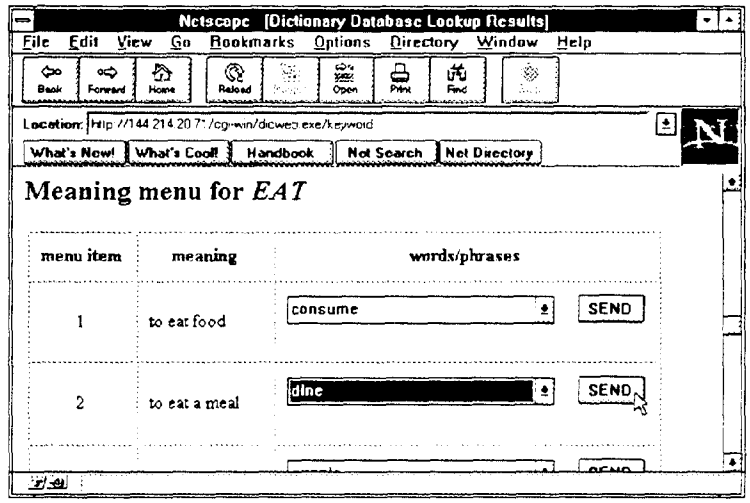


Figure 6: Meaning Menu for *EAT*

Conclusion

The application I have described in this paper is experimental only. No attempt has been made to enter all the information contained in the LLA. I was primarily interested in demonstrating the worthwhile potential of rendering LLA's unique approach to organizing and retrieving dictionary data in the form of a hypermedia presentation for easy web access.

This application has been implemented in a Windows 3.11 environment using 16-bit Visual Basic 3.0 and Access 2. The web server is Denny's 16-bit WinHTTPD. The availability, however, of more advanced web servers for Win95 and Windows NT environments and recent advances in Windows programming, including the release of Visual Basic 4.0 and the new 32-bit JET 3.0 database engine, have transformed VB/Access CGI programming into a realistic alternative to CGI programming for UNIX.

References

Denny, Robert. 1995. VB/Access CGI Programming. http://website.ora.com/devcorner/db-src/index.html

_____. 1995. Visual Basic Sources. http://website.ora.com/devcorner/db-src/vb-top.html

Heslop, B.and L. Rudnick. 1995. HTML Publishing on the Internet. Ventana Press.

Longman Language Activator. 1993. Longman Group.