# Xara: an XML aware tool for corpus searching

Lou Burnard and Tony Dodd

Research Technologies Service,
Oxford University Computing Services
13 Banbury Road, Oxford OX2 6NN
lou.burnard@oucs.ox.ac.uk
tony.dodd@btinternet.com
tel: +44 1865 273285
fax: +44 1865 273275

**From SARA to Xara**

Xara is the working name for a new version of SARA, the `SGML aware retrieval application' originally developed for use with the British National Corpus (BNC) in 1994. The system has been completely rewritten as a general purpose tool for searching large XML corpora, with a particular focus on the needs of corpus linguists, with close attention to new XML-based encoding standards, and with the benefit of hindsight derived from a decade of feedback from hundreds of SARA-users world wide.

The Xara system combines the following components: (1) an *indexer*, which creates inverted file style indexes to a large collection of discrete XML documents; (2) a *server*, which handles all interaction between the client programs and the data files; (3) a Windows *client*, which handles interaction between the server and the user. The modularity of this architecture has several advantages, permitting, for example, the development of multiple specialized client programs for different applications or styles of usage.

In addition, an index building utility, called Indextools, is supplied with Xara, which simplifies the process of constructing a Xara database. Its chief function is to collect information about the corpus to be supplied additional to that present in any pre-existing corpus header, and to produce a validated and extended form of the corpus header. It can also be used to run the indexer and test its output.

Rather than review extensively its history and design philosophy in this short presentation, we give here some general comments covering the following aspects of the system:

- XML support
- Corpus related features
- Ease of use

**XML support**

Xara will process any XML encoded corpus. The more detail present in the tagging, the more facilities are available to the client but the minimal requirement is only that the text be well-formed XML. If the corpus to be processed specifies a document type definition (DTD), then Xara will validate it at indexing time, and will not proceed if any validity errors are discovered. Unlike earlier versions of the program, any DTD may be used for this purpose, though Xara was (naturally) designed with the likely needs of such widely used DTDs as the TEI or CES in mind. Also unlike earlier versions of the program, as previously noted, no DTD at all need be supplied.

Xara can thus do something useful with the full range of digital material one might wish to build into a corpus. At one extreme, we will demonstrate how it can be used to provide basic searching facilities for a collection of Project Gutenberg style texts, innocent of any explicit descriptive markup at all; at the other, we will show how it can also take full advantage of the rich annotation present in a multilingual corpus produced in full conformance with the XCES Guidelines, containing detailed feature structure analysis, POS-tagging, and explicit lemmatization. Oddly enough, the most problematic material is likely to be texts which have been marked up in loose (syntactically incorrect) HTML, such as that

generated by automatic conversion from Microsoft Word; fortunately, utilities such as Dave Ragget's *Tidy* are readily available to generate well-formed XML from such conversions.

With XML support, comes Unicode support. Xara uses Unicode internally to represent all character data: it can thus handle text in any language, and any combination of languages. To take full advantage of this, the user of the system needs convenient methods both for displaying and for inserting Unicode characters at their workstation.

Good Unicode fonts are now available for display of texts in several different scripts (we have so far tested Chinese, Eastern European, Medieval, and Ancient Greek scripts) and the number continues to grow. Xara does not assume the existence of any particular font, however, and allows the user to select the display font at run time. In common with other XML systems, the system will correctly process character entity references found in the data, such references being retained untranslated in the underlying corpus index, but rendered as the appropriate Unicode code point when displayed in non-XML modes.

To enter characters not found on the keyboard, for example to search for words containing them, however, an appropriately configurable input system is necessary. Dynamic keyboard redefinition is built into more recent operating systems, but is still a little user-unfriendly in most cases; the Xara client therefore includes its own keyboard redefinition facilities, which allow selection of specific characters from a Unicode table by point and click, temporary mapping of keyboard keys, or complete redefinition of a new keyboard map, which can be loaded as needed.

**Features for Corpus Linguists**

Any XML or SGML aware search engine has the ability to locate specific tagged components and to carry out searches within the context of such components. Most also have the ability to reorganize and display search results in a variety of forms. SARA extended these facilities with some additional, more lexically-motivated, abilities. These include:

- implicit or explicit tokenization of element content
- implicit or explicit lemmatization of element content
- multiple keys for index searching
- expandable automatic collocation search

Xara supports the full range of facilities originally provided by Sara, but with several modifications and simplifications to the interface. A number of new facilities have also been added.

Xara inherits from SARA a rich range of query facilities. The user can search for substrings or regexp-style patterns, words, phrases, or the tags which delimit XML elements and their descriptive attributes, either simply checking for their presence or absence in the lexical index maintained for the corpus (and inspecting their frequency or collocative patterns therein), or retrieving and displaying actual instances of the words etc. sought for.

Searches can be made using additional keys such as part of speech, or root form (lemma) of a token, specified either explicitly in the tagging of the texts, or implicitly by means of a named algorithm. Xara also supports a variety of scoped queries, searching for combinations of words etc. in particular contexts, which may be defined as XML elements, or as combinations of other identifiable entities, or as stretches of text. Such searches may be order-sensitive or insensitive.

Different kinds of search can be combined to form complex queries of various kinds, using either a simple graphical interface, or a (rather esoteric) `Corpus Query Language' (CQL). This language defines the full capability of the query interface and forms a major part of the protocol by means of which client and server modules communicate. In Xara, it has been re-expressed using an XML syntax, in line with the desire to leverage XML standards wherever possible. The client also supports a simple ECMAscript-like scripting language which makes direct calls on an API defined in very similar terms to the CQL.

Like its predecessor, Xara displays the results of searches either one at a time or within a traditional KWIC style window which can be sorted, thinned, expanded etc. The context of hits located in this way can be explored by expanding it, up to the full text level if necessary. Unlike its predecessor, Xara allows user-definition of a range of formatting properties for KWIC and single hit displays, using a subset of the formatting properties defined by the W3C standard Cascading Stylesheets (CSS); it also allows the user to export results in a simple XML format which can then be reprocessed, either by Xara, or by any other XML-competent application, such as a word processor or XSLT engine.

Corpora can be reorganized or partitioned in a user-defined way, using the results of any query, the values of specified element/attribute combinations, or a manual classification. Searches carried out across partitioned corpora can be analysed by partition: so the client can display the relative frequencies of a given lexical phenomenon in texts of different categories identified in a corpus.

**Using Xara**

For existing users of SARA, the main new feature of Xara will probably be the facilities it offers for indexing of new corpora. As noted above, all the parameters required by the indexer and server are now gathered and stored transparently within a TEI-conformant XML header file, rather than being supplied at runtime by various obscure control files, command line options, or hard-coded declarations. A new Windows utility has been included to facilitate creation and modification of this data: it allows the user to control the behaviour of the indexer by selecting available options from a series of dialogues, and saves the results of these decisions in an appropriate part of the TEI/XCES header. The same utility can be used to check validity of the supplied corpus files and to run the indexing utility, and provides an interface for testing that the system index files have been correctly generated. Its use is not however essential: other XML-aware software can be used to define a corpus header which Xara will use, and the indexer utility can be run independently.

The indexer needs to be provided with the following information:

- how PCDATA (element content) is to be tokenized
- how tokens are to be mapped to index terms (for example, by lemmatization or by the inclusion of additional keys)
- how indexed terms are to be referenced in terms of the document structure
- how and whether XML tags and attributes are to be indexed

Much, perhaps most, of this information is implicit in the XML structure for a richly tagged corpus: one could imagine a corpus in which every index term was explicitly tagged, with lemma, part of speech, representation etc. In practice however, such richly tagged corpora remain imaginary, and software performs the largely automatic task of adding value to a document marked up with only a basic XML structure. The new indextools utility specifies how the task is to be performed in a standardized way, using the existing structure of the TEI header (see http://www.tei-c.org/Guidelines/HD.htm) to hold the specification. Because our original design goal was to avoid any need for extension or modification of the existing TEI DTD; we have used very general constructions of the type`<list type="foo"/>` rather than defining more `application focussed' tags such as `<foo>`. We have however done this is a fairly consistent and easily validated form, so that when TEI support for external namespaces is available, it will be easy to make our generic tags more specific.

The Xara system is currently under beta-test and will be demonstrated at the conference. We hope to make an initial general release available by the summer of 2003, but participants wishing to experiment with it earlier are very welcome to participate in the beta test. The first release will operate in any modern Microsoft Windows environment; it is intended, however, to port both indexer and server to Unix environments as soon as possible; it is also intended to licence open source versions of at least those components of the system at that time (open source licensing of the Microsoft-dependent components is rather less likely, for obvious reasons)