FQS FORUM: QUALITATIVE SOCIAL RESEARCH SOZIALFORSCHUNG

Volume 3, No. 2, Art. 13
May 2002

Extensible Markup Language and Qualitative Data Analysis

Patrick Carmichael

Key words: CAQDAS, computer-aided qualitative data analysis, XML, WWW, Internet, networking, client-server, information architecture, metadata

Abstract: The increasing popularity of Extensible Markup Language (XML) and the availability of software capable of reading and editing XML documents present opportunities for Qualitative Data Analysis (QDA) facilities to be incorporated into "groupware" applications such as collaborative workspaces and "document bases", and to be made available across networks both within organisations and across the Internet. Collaborative systems have, in the past, characteristically, been geared to retrieve and present whole documents, and while annotation and discussion of documents has been possible within such systems, the "pencil-level" analysis commonplace in CAQDAS (Computer Assisted Qualitative Data Analysis Software) has been lacking. XML, when combined with a scripting language such as Perl, can be used to offer basic QDA functionality—retrieval by text and codes, attachment of memos to text fragments, and the generation of summary data—via a standard web-browser.

Table of Contents

1. Introduction

Within the School of Education at the University of Reading, staff members and students use a variety of qualitative data analysis software (principally Nudist, The Ethnograph and Inspiration) and collaborative networking tools in support of research projects, school partnerships and World Wide Web resource development. [1]

Particular use has been made of BSCW (Basic Support for Collaborative Work), a "groupware" application designed to allow users across a network to disseminate and discuss a wide range of documents including, but not restricted, to web pages, images and Microsoft "Office" materials (BENTLEY et al., 1997). Within the School of Education, this application has been used for document dissemination; for collaborative authoring and peer-review of documents; and for discussion of conference papers by those unable to attend the "real world" conference at which they were presented. While the BSCW application allows "discussion threads" to be appended to any document, a recent review of BSCW use states that "the vast majority of the user actions consists of browsing through BSCW workspaces and reading documents, comments on documents, notes, etc." (APPELT, 2001, p.5). [2]

Recently, a number of projects have been undertaken for which BSCW, with its "document-level" functionality, has been unsuitable. One of these involves collaboration with Survivors' Fund (http://www.survivors-fund.org.uk), a UK-based non-governmental organisation working in partnership with organisations and projects in Rwanda, most notably AVEGA (*Association des Veuves du Genocide d'Avril*) in order to produce a database of accounts of survivors of the Rwandan Genocide of 1994. Rather than comprising structured interview data after the model of African Rights study (1995), which still remains the authoritative account of the genocide, or the guidelines issued by the United States Holocaust Memorial Museum (1998) these accounts are largely unstructured narratives collected during non-directive interviews and are in some cases fragmentary, some of the respondents having been children in 1994. These accounts complement those in earlier collections, and illuminate and extend the information in other published resources including Human Rights Watch's survey of violence against women during the genocide (1996). While there have been a number of accounts written describing and analysing the events of 1994 (KEANE, 1996; MELVERN, 2000), and historical accounts placing the genocide of 1994 in historical context (PRUNIER, 1995), this expanding body of data provides evidence of the experience of survivors of genocide against the background of continuing social upheaval, instability and politically-inspired violence described by GOUREVITCH (1998). [3]

A second area in which it has proved necessary to provide "finer-grained" analytical tools is in support of schoolteachers undertaking classroom-based research, either individually or as part of larger research projects. In this latter category, Learning how to Learn is a four-year Economic and Social Research Council (ESRC) funded project concerned with the development and promotion of good classroom practice in relation to student assessment in primary and secondary schools in the UK. While there is already an established research base of materials in this area (BLACK & WILIAM, 1998), this project involves the development of a database of exemplary materials collected in classrooms by teachers and researchers. While the content matter is very different from that in the Genocide Survivors' project, this evidence is once again fragmentary and varies in format. Much of it comprises short transcripts of classroom discourse, "anecdotal" evidence provided by teachers and field notes collected by researchers. [4]

Both the Rwandan Survivors' and school-based projects have university-based staff associated with them; these staff members have access to Qualitative Data Analysis software and this is used to structure data fragments, analyse transcripts and to contribute towards theory-building. In the Genocide Survivors' Project, Nudist has been used and within the Learning to Learn research team, ATLAS.ti is used by the researchers responsible for collection and analysis of interview data. At the same time, both projects have the declared aim of involving as broad a range of "stakeholders" as possible, their role ideally extending beyond contribution of data to playing a role in its interpretation. Whether this involved genocide survivors adding their comments, experiences and local knowledge to existing accounts, or teachers offering their insights into classroom activity, the need was the same: to provide some form of CAQDAS functionality to the widest possible audience—many of whom were distributed across large geographical areas and whose access to IT in general, and to CAQDAS software in particular was likely to be, at best, intermittent. [5]

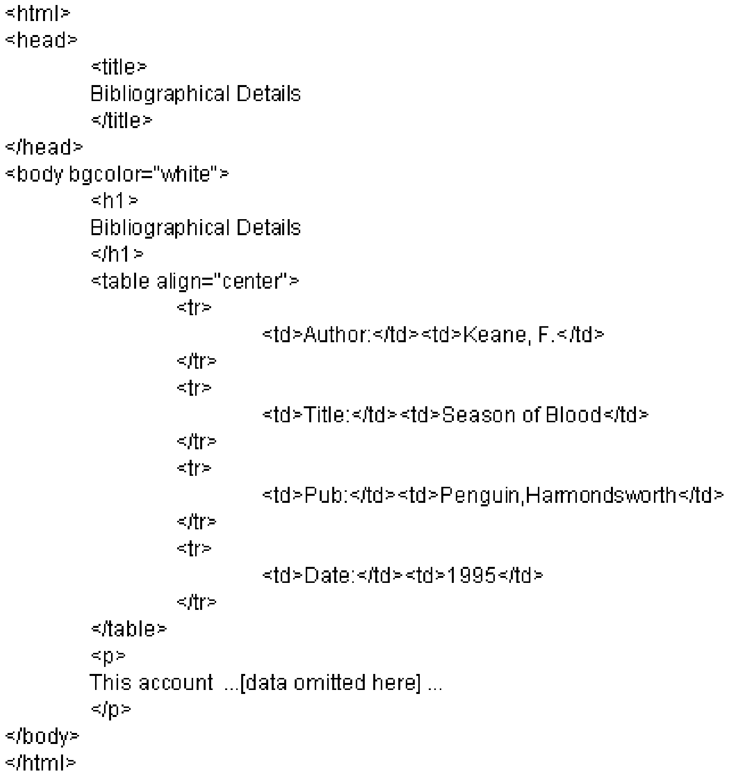
What was needed was a software solution offering the cross-platform, networked features of a groupware application like BSCW, but extended to include at least some Qualitative Data Analysis tools. At the same time, it was felt that currently available CAQDAS software, already recognised as suffering to a greater or lesser extent from problems caused by lack of extensibility and platform-specificity (FIELDING & LEE, 1998, pp.69-71; MUHR, 2000), was inappropriate, since these packages make little use of the "client-server" or "three-tiered" network architectures necessary to provide secure multi-user access to networked data repositories. In both projects, we could not make any assumptions about the hardware and software available to users beyond having intermittent access to the Internet, possibly via public-access terminals. It was against this background that the decision was taken to use XML as the basis of data storage and analysis across these two projects. The examples of XML data structures and processing that are illustrated in the remainder of this paper are drawn from the Rwanda Genocide Survivors project. [6]

2. Extensible Markup Language

Extensible Markup Language (XML) is a development of Standard Generalized Markup Language (SGML) and is currently being used as the basis of many networked applications within enterprises and across the Internet. It differs from the more familiar Hypertext Markup Language (HTML) used for the layout of web pages in that it is primarily for data description rather than for data presentation, and has been characterised as a "metalanguage" (BAEZA-YATES & RIBIERO-NETO, 1999, p.161). GOLDFARB writes:

> "XML data is smart data HTML tells how the data should look, but XML tells you what it means but XML data isn't just smart data, it's also a smart document and you don't have to decide whether your information is data or documents; in XML, it is always both at once You can do data processing or document processing or both at the same time" (GOLDFARB, 2000). [7]

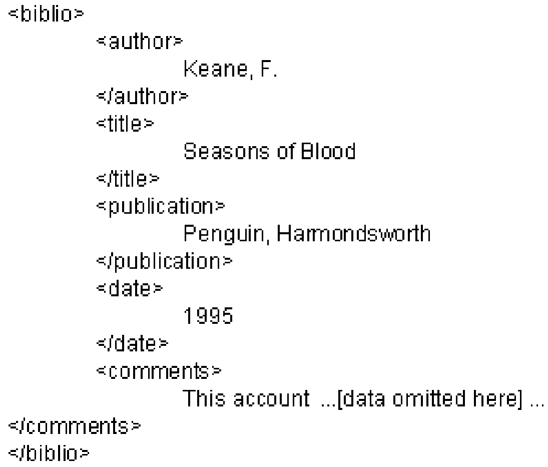
Unlike the HTML used to lay out web pages, XML uses user-defined tags to provide structure and descriptive information about the data. In the following illustrative example, the same data is marked up using HTML and XML. Listing 1 shows some quite straightforward HTML which sets out a table of bibliographical information followed by a paragraph of notes.



```
<html>
<head>
        <title>
        Bibliographical Details
        </title>
</head>
<body bgcolor="white">
        <h1>
        Bibliographical Details
        </h1>
        <table align="center">
                <tr>
                        <td>Author:</td><td>Keane, F.</td>
                </tr>
                <tr>
                        <td>Title:</td><td>Season of Blood</td>
                </tr>
                <tr>
                        <td>Pub:</td><td>Penguin,Harmondsworth</td>
                </tr>
                <tr>
                        <td>Date:</td><td>1995</td>
                </tr>
        </table>
        <p>
        This account  ...[data omitted here] ...
        </p>
</body>
</html>
```

Listing 1: HTML [8]

This HTML (which is rather better structured than that on many web pages), demonstrates some of the inherent problems of using a system designed for layout in order to describe data. Tags are in some cases descriptive (<title>) but in other cases purely concerned with presentation on the page (<table>, <h1>) In some cases data description and presentation information are combined in a single tag (<body bgcolor="white">) While it is possible to insert extra attributes inside tags so that pages can be used as the basis of a text retrieval system (for example <td id="author">Keane, F</td>) it would be more useful if it were possible to remove all tags which are not concerned with data description, as in Listing 2, which shows the same data as XML.



```
<biblio>
        <author>
                Keane, F.
        </author>
        <title>
                Seasons of Blood
        </title>
        <publication>
                Penguin, Harmondsworth
        </publication>
        <date>
                1995
        </date>
        <comments>
                This account  ...[data omitted here] ...
</comments>
</biblio>
```

Listing 2: XML [9]

In this case, the tag names are meaningful and all presentational information has been extracted. Presentation onto a web page might be achieved through the application of a "style sheet", which is used to insert layout tags and would have the benefit of being applicable to any document using the same system of data tags. While some websites do use XML as the basis of their data description and presentation (see, for example, FLOYD, 1999), much work to date has used XML as the basis of server-to-server applications within which web page production plays only a small part. BizTalk, for example, is an initiative designed to support XML business-to-business (B2B) document exchange both within enterprises and over the Internet (http://www.biztalk.org), while the Dublin Core Metadata Initiative (http://purl.org/DC) seeks to provide a system for the consistent description of bibliographical resources within and between libraries and archives. [10]

While many organisations and networks have found the potential for data description offered by XML appealing, its ancestry in the markup languages of the publishing industry (MARCHAL, 1999) has perhaps led to its potential as a tool for qualitative data analysis being overlooked. This is evident in many handbooks and tutorials which stress the importance of developing a Document Type Definition (DTD) which defines the structure of data and documents very precisely, an approach which seems to restrict the extensibility of XML itself and ignores the fact that it offers far more than simply the opportunity for users to define their own set up mark-up tags and codes. [11]

The separation of raw data from subsequent layers of markup—whether interpretative or concerned with formatting and output—means that XML datasets are characteristically subject to what MARCHAL calls "non-destructive transformation" (1999) which leave the source data unchanged. This means that the potential exists for the development of something akin to an "expert system" appropriate to the research domain, in which a corpus of texts and other resources are used as the basis of progressively-elaborated interpretative schemes which can then be used for analysis of new data. True expert systems, particularly those written in the Prolog language, tend to exhibit similar pattern of abstraction of interpretative frameworks from data. BOWEN (1991) describes how "pure" domain knowledge and metadata are combined into "encumbered knowledge" at the "middle tier" of a Prolog expert system. [12]

XML's "unencumbered" nature—with no superfluous application-specific formatting information—makes it ideal as a data exchange format—a fact recognised by MUHR (2000) who describes how the ATLAS.ti system is increasingly using XML: currently, codes and memos can be outputted as XML, and further integration is promised. The promise of XML for those involved in Qualitative Data Analysis is that, as in other areas of networked collaboration across the Internet, "raw" datasets, proprietary software like ATLAS.ti and other project-specific applications can coexist, linked by an underlying common data format. An extension of this functionality, aided by the publication of the "schemata" used to structure documents, would be the physical separation of the original datasets from analytical tools and analyses, as long as the underlying data formats are consistently applied, an XML-based CAQDAS application would, like applications in other fields, be able to run transparently across a network. [13]

In the next sections of this paper I will describe some of the opportunities offered by an XML-based approach to data analysis within the contexts of the two projects already described. The examples provided are, in some cases concerned with replicating the features of existing CAQDAS packages: but differ from them in that they are designed to be used across networks such as the Internet. What is, however, evident, is the fact that many of the features of existing proprietary packages can be replicated using XML-based solutions, while, at the same time, its flexibility and potential as a data exchange format allows it to be used in applications which transcend the descriptive frameworks used by WEITZMAN and MILES (1995) and RICHARDS and RICHARDS (1994) to describe CAQDAS packages. This is a particularly valuable feature in the context of long-running projects. [14]

3. XML and Varieties of Qualitative Data Analysis Software

WEITZMAN and MILES (1995) and FIELDING and LEE (1998) all distinguish between "generic" software which can be used in qualitative data analysis and dedicated software packages RICHARDS and RICHARDS (1994) provide a similar framework, although there are some differences in their categorisations. Generic software includes word processors, text retrieval systems and textbase managers while dedicated QDA software are categorised as "code and retrieve" systems (such as The Ethnograph), code-based theory-builders (such as Nudist and ATLAS.ti) and conceptual network builders including Inspiration. To this can be added programs which are being used out of their intended context; examples include process design and project management software such as Visio and Microsoft Project which offer good conceptual mapping and "timeline" tools respectively. It is recognised, however, that some software may be used in more than one of these modes (WEITZMAN & MILES, 1995) and also that many users do not fully utilise all the functions of the software available. [15]

The appeal of a data storage and analysis system based around XML is that these distinctions are rendered largely redundant. A collection of data stored as XML can be used for simple text retrieval; can be extended into a code-and-retrieve system in which data is retrieved on the basis of code content; and can ultimately be developed into a conceptual mapping tool in which data is presented in either a structured-text or graphical form using a data visualisation tools. This versatility has already been recognised in other areas: QUIN describes an exemplary network-enabled application called "BookWeb" (2000) which demonstrates the processes involved in coding, indexing and retrieving for display a range of XML data, and demonstrates how an initial data set (in this case, consisting of bibliographical information) can be used as the basis of a "free-text" search and retrieve application, a keyword and retrieve application or can be transformed into graphical representations such as conceptual maps. [16]

Within the projects I have described, with their diverse audiences, the capabilities both to offer different types of analytical tools and to develop those tools as the projects develop have been identified as being of value. For example, within "Learning to Learn", one of the types of materials to be presented via web pages to teachers and researchers is transcripts of teachers' and learners' classroom discourse. For some users, the capability to search the database and retrieve examples relevant to their own practice—perhaps according to the age of learners or the curriculum area being addressed—will be enough. Others will want to be able to retrieve examples on the basis of codes added by the originating teacher, perhaps illuminating the transcript, or explaining the rationale for the application of a particular teaching strategy. As the project progresses, however, users will be encouraged to add their own assessments and interpretations, necessitating a move beyond simple data retrieval. The university-based research team plans to undertake further levels of analysis; in this case, the database comprises not only the original classroom data, but also the interpretations placed upon them by the teacher-researchers. The XML-based approach adopted in the project allows this progression to take place without the need for users to adopt new software. [17]

4. The Structure of an XML-based Application

Most XML-based applications are "three-tiered" networked applications, the user gaining access to the XML "data tier" through a "client tier" interface such as a web browser, a "console" application or some other dedicated program. The vast majority of XML applications are server-based and accessed through web browsers such as Netscape Navigator or Internet Explorer, necessitating a conversion at some point from XML into HTML. This conversion take place in the "middle tier", which is also the location of search and retrieve programs, libraries of codes, "style sheets" providing information about how to present output, and most critically, a "parser" written in one of a number of programming languages (see Figure 1).

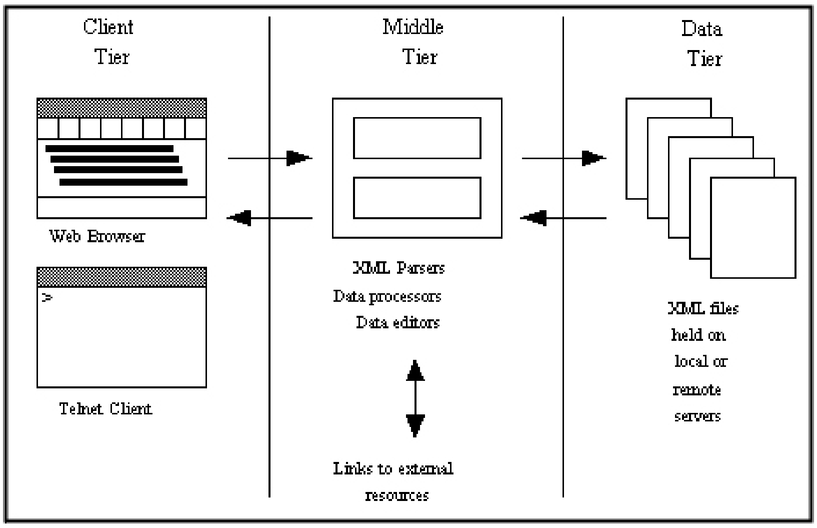
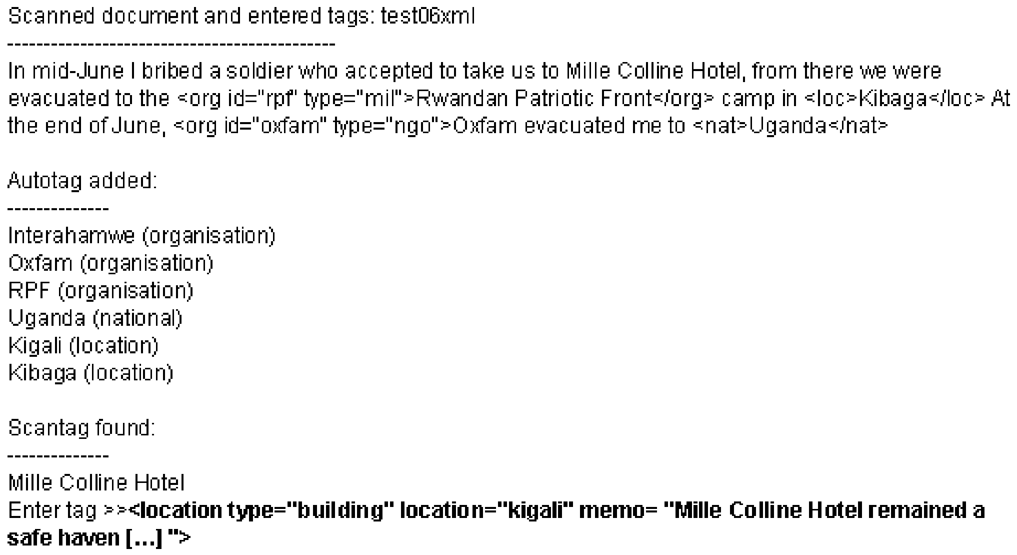


Figure 1: Three-tiered architecture showing how the client accesses XML data via a "middle tier" [18]

MARCHAL (1999) stresses the importance of establishing the right level of what he describes as "abstraction": the extent to which information is included in the middle tier rather than in data itself. Within a three-tiered XML application, for example, codes attached to text fragments could either be written into the source data (not recommended as this represents a "destructive transformation" of the original data); written into a new version of the original document which could then be assigned a new and unique identification; or stored externally on the "middle tier". [19]

While the second of these approaches is the most flexible, the last approach—the development of external "libraries" has some value when the originators of the database wish to flag up key data elements for less experienced users viewing content across a network. This was the case in the Rwanda Genocide Survivors' database. While this data was collected during non-directive interviews, interviewers were asked to prompt interviewees for specific information relating to locations, organisations, names and dates wherever possible. Lists of these were included in an external library so that, should they appear in any document in the database, they could automatically have codes attached. A simple "command-line" administration interface was used to scan new documents for potential "library entries" on the basis of capitalisation and their non-appearance in a dictionary (see Listing 3).



```
Scanned document and entered tags: test06xml
---------------------------------------------
In mid-June I bribed a soldier who accepted to take us to Mille Colline Hotel, from there we were
evacuated to the <org id="rpf" type="mil">Rwandan Patriotic Front</org> camp in <loc>Kibaga</loc> At
the end of June, <org id="oxfam" type="ngo">Oxfam evacuated me to <nat>Uganda</nat>

Autotag added:
--------------
Interahamwe (organisation)
Oxfam (organisation)
RPF (organisation)
Uganda (national)
Kigali (location)
Kibaga (location)

Scantag found:
--------------
Mille Colline Hotel
Enter tag >><location type="building" location="kigali" memo= "Mille Colline Hotel remained a
safe haven [...] ">
```

Listing 3: Scanning for known Text Strings with Perl (output of "Autotag" and "Scantag" utilities) [20]

This "middle-tier" feature and most of the others described here, make use of the Perl programming language. While other languages (Java, Python, Visual Basic and PHP, for example) offer support for XML processing, Perl also offers the most extensive and well-integrated "regular expression" syntax for pattern searching of any language (FRIEDL, 1997) and a range of useful add-on modules and interfaces to other software. While it is eclectic in style, Perl is described by its creator as "a language for getting things done" (WALL, CHRISTIANSEN & SCHWARTZ, 1996, p.ix). In my view, Perl has considerable potential as the basis of any web-based Qualitative Data Analysis application because of its capability to parse, index, manipulate and summarise text. CHRISTIANSEN and TORKINGTON (1998, pp.218-220) set out a list of useful "regular expressions" such as those for "finding initial-caps words" and "extracting sentences" together with more complex examples such as "extracting a range of lines". [21]

Another example of Perl's use relates to the handling of dates. Many of the Genocide survivors' accounts make reference to specific dates and periods such as the assassination of President Habayarimana on 6th April 1994 and the period of French military deployment in the south of the country (Operation Turquoise). Others, particularly those of children, are more vague and make references to sequences of events without providing specific dates. When coding accounts, dates were converted to ISO (International Standards Organization) format using an existing Perl module called "Date::Manip" (http://wwwcpanorg/modules/by-authors/Sullivan_Beck/DateManip-5.38.tar.gz; broken link, September 2002, FQS). This "converts strings like '2 weeks ago Friday' and '2nd Sunday in 1996' and returns the decoded date" (CHRISTIANSEN & TORKINGTON, 1998). Thus "the sixth of April" would be coded as:

```
<date strictdate="1994-04-06">the sixth of April</date>
```

allowing searches across a database for references to specific dates or timeframes. Other relevant Perl modules and interfaces include "String::Approx" which allows approximate matching of text strings and "Lingua::Wordnet" (BRIAN, 2000) which provides an interface to the Wordnet lexical database (http://www.cogsci.princeton.edu/~wn/), in which nouns, verbs, adjectives and adverbs are organised into synonym sets, each representing one underlying lexical concept (FELLBAUM, 1998). [22]

The suite of "middle-tier" features described here allowed the development of an effective data retrieval system operating through a web browser, with a Perl program generating web pages from the XML, converting tags containing codes into hypertext links to further information or "pop-up" labels. In addition, a search facility allowed users to retrieve documents or document fragments, different regular expressions allowing the user to set the target or targets of their searches; their context; the amount of that context to be displayed; and the mode of presentation. Regular expressions can be used simply to locate targets, to change their appearance within the original text when it is outputted to the client (by emboldening, for example), to produce an elaborated version of the original with tags and annotation, or to alter permanently the original text by inserting a tag or annotation. [23]

Regular expression support already exists both in a range of generic and dedicated QDA software, but that offered by Perl, particularly when combined with other Perl functions and modules is particularly rich, and it has proved relatively easy to develop the initial text-retrieval system into a "code-and-retrieve" one in which users can retrieve data on the basis of text matching, attached codes or combinations of the two. [24]

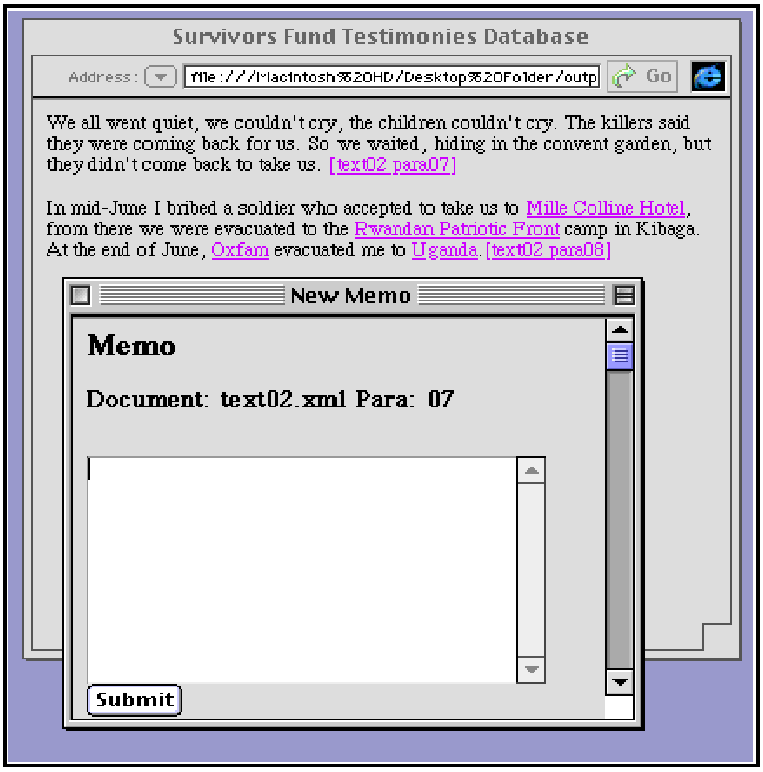
At the same time, regular expression programming is potentially complex and developing robust Perl regular expressions can involve a considerable amount of work. FRIEDL's exemplary regular expression (1997) for matching all valid email addresses is over 6000 characters long; this is why most networked searches take place via a web browser interface, which shields the user from the regular expression actually doing the work. Within both of the projects described here, the size and scope of the databases made it worth investing the time in the development of a customised search tool, optimised for the data and coding systems to be used, but this aspect of development might represent a major hurdle for smaller and shorter-term projects. [25]

5. From Automatic Coding to Interactive Memoing

At this stage in the development of our project databases and the analysis tools to be used with them, it became clear that there were two major deficiencies in the approach. Firstly, while, it might seem that the power of regular expressions makes them a sufficient basis for text transformation and retrieval (albeit through some kind of client interface) changes in the structure or content of data may necessitate updating of search mechanisms—which is a non-trivial task given the complexity of regular expression syntax. Secondly, and perhaps more critically, coding was largely achieved through a non-interactive interface. While this proved useful in producing output with hypertext facilities, it still resulted in a largely asymmetrical system in which most users had access only to a small range of the functionality common in proprietary CAQDAS packages. The largely automatic coding processes effected by regular-expression substitution insertion of coding is no substitute for the more detailed "memoing" necessary to develop interpretational frameworks and which contributes to effective theory-building (GLASER, 1978, pp.83ff.). [26]

With further development, "tiered" XML applications could offer users the opportunity to interactively code or add "memos" to data—either altering the original data in the process or building their own version of it elaborated with their own comments. As before, a number of solutions already exist, but they have yet to be applied to networked Qualitative Data Analysis applications. UDELL (1999, p.207) describes a "reviewable document base" in which Perl regular expressions are used to insert hypertext links into text documents, making each title and paragraph a point to which memos can be attached. [27]

This technique has been used on the Genocide survivors' accounts, within which each title, paragraph and other coded data fragment serves as the "anchor" or departure point for a memo. When viewed through a web browser, the memo "anchors" within the text show as hypertext links which, when clicked, generate a small Javascript window in which the memo can be written prior to being posted to the data tier either as plain text or XML, where it can be appended to the data or stored in an external file (see Figure 2).


Survivors Fund Testimonies Database
Address:
file://Macintosh%20HD/Desktop%20Folder/outp
Go
New Memo
Submit

We all went quiet, we couldn't cry, the children couldn't cry. The killers said they were coming back for us. So we waited, hiding in the convent garden, but they didn't come back to take us. [text02 para07]

In mid-June I bribed a soldier who accepted to take us to Mille Colline Hotel, from there we were evacuated to the Rwandan Patriotic Front camp in Kibaga. At the end of June, Oxfam evacuated me to Uganda. [text02 para08]

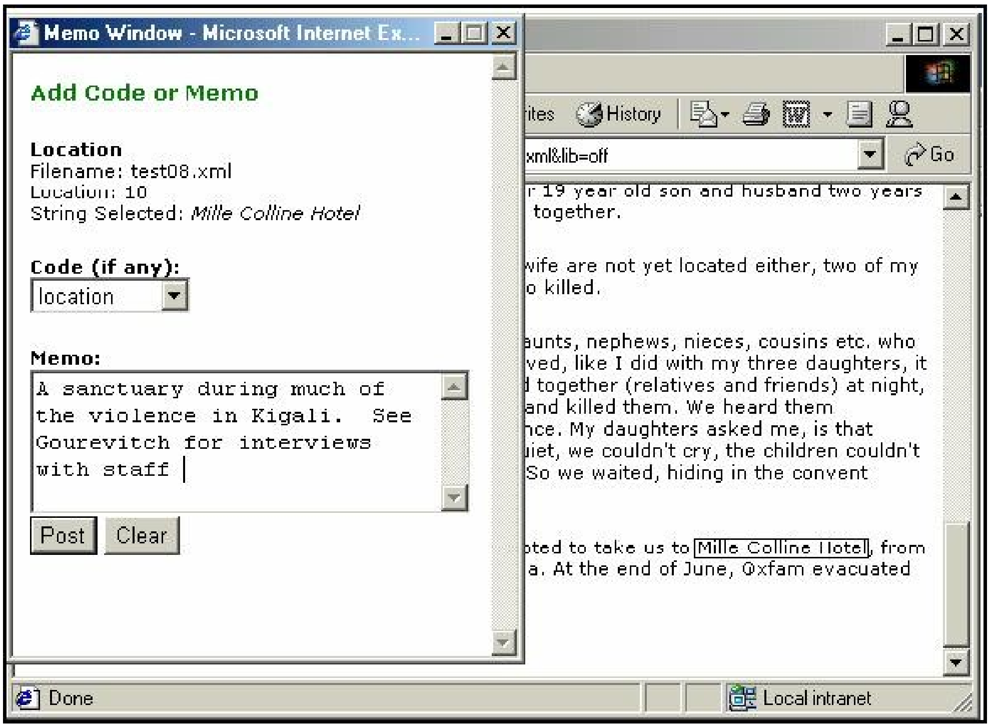
Memo

Document: text02.xml Para: 07

Figure 2: Web Interface showing dynamically generated Memo window [28]

One of the areas in which any web-based system (XML-based or otherwise) may have difficulty competing with "complete" proprietary packages is in providing the "pencil-level richness" demanded by WEITZMAN and MILES (1995, p.335)—specifically in the provision of the capability to highlight small "chunks" of data and add codes or memos. The solution proposed above—memoing and hyperlinking at paragraph level—may not provide sufficient control for many researchers. Such richness demands an interactive user interface and until recently, this has proved difficult to implement via web browsers (although see THOENY, 2000, for a description of Twiki, a collaborative web environment offering read/write web pages aimed at collaborative authoring projects). [29]

With the latest generation of web browsers, and the development of DHTML (Dynamic Hypertext Markup Language), it has become possible to annotate and rewrite web pages more interactively. In particular, the addition of "selection" and "textrange" objects to the object model for web pages, together with "mouse-capture" (in which the position of the cursor over the page is recorded and can be accessed programmatically) makes it possible to develop web pages on which coding and memoing can be based. FRANCIS, HOMER and ULLMAN (1999) discuss these and other features of DHTML in detail. One restriction placed on such an interface is that it cannot directly access source files—the middle tier is still required—so an intermediate "holding area" on the client must be used for all annotations prior to their being posted and verified at the middle tier before the source files are altered in any way. Such an interface, now being used as the basis for coding and memoing of data from the Rwandan survivors, is illustrated in Figure 3. When any text fragment is highlighted, a pop-up window (scripted in Visual Basic Script) appears. This inherits from the middle tier a list of codes currently in use, allowing the appropriate one to be selected and a memo to be added. When this is submitted, the memo itself is stored as a small XML data fragment which can be viewed alongside the source data in a number of formats (see Figure 4).



Add Code or Memo

Location
Filename: test08.xml
Location: 10
String Selected: *Mille Colline Hotel*

Code (if any):
location

Memo:
A sanctuary during much of the violence in Kigali. See Gourevitch for interviews with staff

xml&lib=off

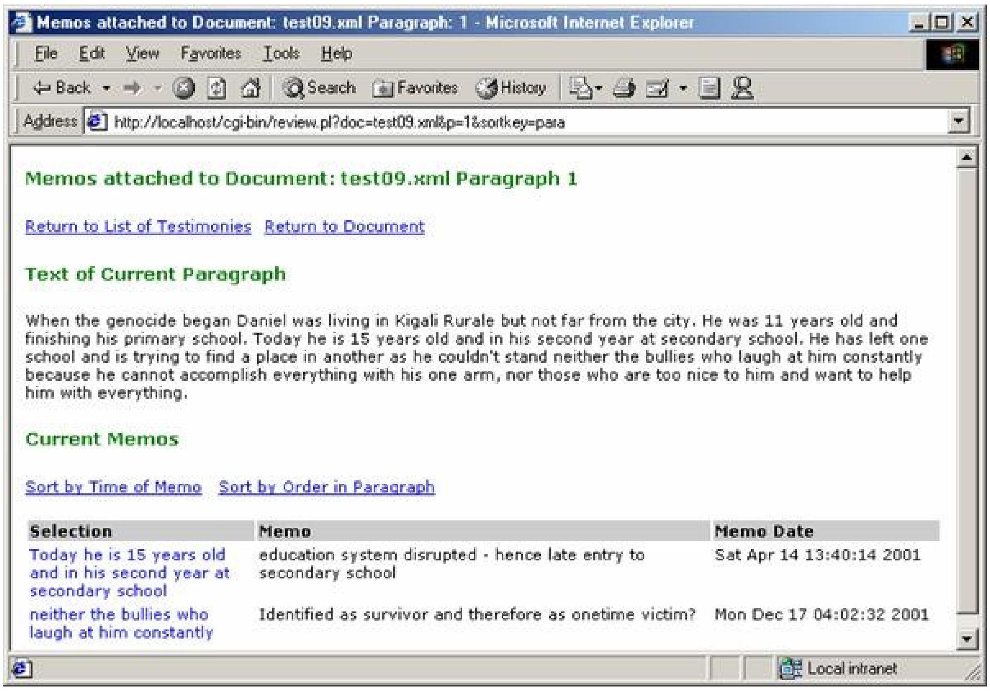
r 19 year old son and husband two years
together.

wife are not yet located either, two of my
o killed.

aunts, nephews, nieces, cousins etc. who
ved, like I did with my three daughters, it
l together (relatives and friends) at night,
and killed them. We heard them
nce. My daughters asked me, is that
uiet, we couldn't cry, the children couldn't
So we waited, hiding in the convent

pted to take us to Mille Colline Hotel, from
a. At the end of June, Oxfam evacuated

Figure 3: Using "selection" of text to capture data from web browser display. The selection is incorporated into the newly generated "New Code/Memo" window. Selection, location, code and memo are then "submitted" to the middle tier for incorporation into the data tier. [30]


Memos attached to Document: test09.xml Paragraph: 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/review.pl?doc=test09.xml&p=1&sortkey=para
Local intranet

Memos attached to Document: test09.xml Paragraph 1

Return to List of Testimonies Return to Document

Text of Current Paragraph

When the genocide began Daniel was living in Kigali Rurale but not far from the city. He was 11 years old and finishing his primary school. Today he is 15 years old and in his second year at secondary school. He has left one school and is trying to find a place in another as he couldn't stand neither the bullies who laugh at him constantly because he cannot accomplish everything with his one arm, nor those who are too nice to him and want to help him with everything.

Current Memos

Sort by Time of Memo Sort by Order in Paragraph

Selection	Memo	Memo Date
Today he is 15 years old and in his second year at secondary school	education system disrupted - hence late entry to secondary school	Sat Apr 14 13:40:14 2001
neither the bullies who laugh at him constantly	Identified as survivor and therefore as onetime victim?	Mon Dec 17 04:02:32 2001

Figure 4: Memos added to specific paragraphs via the "pop-up" window are displayed. The text of the paragraph can also be displayed with all coded fragments highlighted. [31]

Even if it were not used for entire projects, this could well provide a useful tool in the early stages of research during which researchers "double-code" accounts in order to negotiate reliable sets of codes. More complex web interfaces of this sort also offer considerable opportunities for supporting and presenting the results of the "parallel annotation" characteristic of collaborative and action research projects (WEITZMAN & MILES, 1995). [32]

One extension of this approach is the presentation of these memos as a threaded "conference" or "discussion forum" in which different researchers could compare their interpretations with those of others or take part in discussions with other researchers, observers or even with the subjects of their research. The recent launch of XMLBoard, (http://hypermartxmlboardnet; broken link, September 2002, FQS), a web-based "discussion forum" application which stores data entirely in XML makes it possible for memos and ensuing discussions to be integrated with an existing set of XML documents. The consistent language of description allows the researcher to request, via the middle tier, a document and all its attached memos; all memos attached to specific library elements; or all memos themselves containing specific text. The combination of XML and dynamic generation of hypertext links—to memos, discussions, code libraries and other external resources, may make it possible for hypertexts, previously regarded as difficult to develop and labour-intensive (CORDINGLEY, 1991; FISCHER, 1994) to play a greater role in qualitative data analysis in general. Furthermore, it allows the integration of QDA techniques into more general groupware applications, making them more appropriate for use in fields such as education where they can be used to support "teacher-as-researcher" projects. [33]

6. XML Parsers as Generic Data Retrieval Tools

Most retrieval of data formatted (manually or automatically) takes place via a middle- tier application known as a parser, which has built-in procedures for handling tags and the data between them. MACHERIUS, RODRIGUES, DERKSEN, ALMEIDA and JOSTEN (2000) warn against the use of regular expressions for any but the simplest text retrieval tasks and review the range of Perl XML parsers available. Broadly, parsers are either stream-based, reading through data sequentially without building an in-memory representation, the appearance of specific tags causing procedures to search, extract or alter the text within them to be invoked; or tree-based, in which the entire document is read into memory so that it may then be "browsed" up and down by the parser. The former are faster and make fewer demands on system resources, while the latter make the data structures easier to navigate and extract. [34]

RODRIGUES' "hybrid" Perl parser "XML::Twig" was used This parser is a development of the generic stream-based Perl::XML parser but works in stream mode only until it finds specified tags, at which point it builds a small tree (the "twig") in memory for that part of the document (RODRIGUES, 2000). This makes it ideal for handling large sets of data of which only small sections may be of interest to the user at any time. The parser can act as a "wrapper" for a range of text-manipulation and retrieval procedures, which may be called as and when appropriate. For example, an account might be parsed and the XML tags might be mapped to appropriate HTML tags so that it can be displayed "read-only" in a web browser; alternatively, a user request for accounts mentioning events in Kigali from July 1994 onwards could rapidly parse a large number of documents in stream mode returning only those paragraphs (labelled as to their document of origin) with dates and locations highlighted. [35]

A parser, as a middle-tier application, is also able to access the full range of researcher memos, libraries of entities and external resources such as lexical databases which have been built up as a part of the enquiry process and to output the results to the client. Most critically, parsers are sufficiently robust that they can read any document which is syntactically correct XML. This allows them to used across networks to process any data conforming—in terms of format and code-set—to project definitions. Parsers could also be set up to generate a report of the current use of tags—both those inserted manually by the researcher and those generated automatically. This is possible because when parsers are moving through XML documents, they can be set to return either the tags and their attributes (information held inside the tag) or the data between the opening and closing tags, or some combination of these. Printing out all tags together with information about their application equates with allowing the printing out of the current "code book" providing a basis of potential collaboration between researchers. [36]

XML can provide the basis of theory-testing enquiries, with progressive "runs" of the parser being used to explore different patterns within data and codes. Because of XML's lack of any inbuilt data presentation information, the parser can also be set up to convert "raw" data to other formats including Comma-Separated Values (CSV), for import into Excel or SPSS; into different kinds of web resources from interactive discussion forums to "read-only" web pages; into plain text, bereft of any tags or interpretations whatsoever; or simply as XML with the basic layout tags and information about the source and context of the data collection (CROSS, 2001). [37]

The capability to remove all codes and return data to their "unencumbered" form is important. As FIELDING and LEE state, "concerns might arise about how conclusions have been reached; in other words, the question is 'Have research data been misrepresented?'" (1998, p.68). If one of the options open to the research audience is to access the "raw" data, then a degree of transparency is provided which is often difficult to achieve when using computer-based QDA tools. Transparency, as FIELDING and LEE (1998) point out also involves exposure of research processes to scrutiny and this may mean that middleware applications should be made available to interested parties. Perl is an "open source" software (meaning that source code is freely available) and XML is a published standard of the World Wide Web Consortium (http://www.w3.org). [38]

7. Opportunities for Further Development

The facilities described here have allowed the two projects to offer basic QDA facilities to a widely-dispersed audience of potential contributors to a collaborative research enterprise, while at the same time maintaining the integrity of expanding server-side databases. The choice of XML as the means of data description will allow these two projects, both of which have potentially long lives, to expand according to the needs of researchers and "clients" of various types. That said, there are areas in which future technological developments are likely to take place. The following are specifically under our consideration at present:

- One of the main areas which needs to be addressed is MARCHAL's question about the "level of abstraction" of metadata (codes and memos) from the data to which they refer, and to review the implications of dispersing these elements across a network. A related issue is the need for XML documents to be consistently structured in order for parsers to read them accurately (or at all). Since XML syntax does not allow overlapping tags (all tag pairs, the opening one of which contains coding information, have to be "nested"), controls will need to be developed in order that data which is being coded and annotated by groups of researchers maintains its integrity.
- Even with the retrieval capabilities offered by Perl's regular expressions, complex queries are difficult to construct; the appearance of XPath (XML Path Language: http://www.w3.org/TR/xpath) and of parsers which support it may make it easier to construct the kind of compound and complex queries which researchers use in the course of theory-building and testing. Since XPath also supports numerical comparison operators (these have to be built on to standard XML parsers) this should make it easier to construct queries that seek to specify or exclude instances on the basis of numerical or date values.

- Despite developments in client-side web browsers, using these as the basis of "read-write" functionality remains problematic. Since all actions (retrieving, adding codes, adding memos) require a call (via the Common Gateway Interface or CGI) to the middle tier on the server, coding can be a slow process. It also makes data validation on the middle tier essential, since access by URL not only to data, but also to programmatic resources, represents a potential point of incursion into the system. Ultimately, it may be that a dedicated client-side program is developed (possibly in Visual C++) offering those specific aspects of the web browser functionality required by the networked QDA application.
- A final area for development is concerned with what in document-based systems is known as "version control". Across a networked application with many potential contributors and users, even with controls in place to prevent "destructive transformation" of data, there is a danger that different versions of original documents, varying in their level or scheme of coding and annotation, or drawing on different "middle tier libraries" might come into existence. It is thus essential that codes and annotations added to any data include some kind of unique identifier of the researcher (such as a URI), an identification of the coding system or codebook used and any dependencies on external resources. This is addressed to some extent within the XML specification from the World Wide Web consortium by the idea of "namespaces" (BRAY, HOLLANDER and LAYMAN, 1999) but it has proved useful within our projects to require the attachment of explicit "meta-metadata" to this effect within data files and the "registration" and assignment of a unique URI to all researchers involved in data analysis. [39]

Acknowledgements

The author gratefully acknowledges the assistance of members of the Perl programming community who have knowingly or unknowingly provided resources and help in the development of these projects; Mary Kayitesi BLEWITT, coordinator of Survivors' Fund for her help and for providing this opportunity to work on the Rwandan Survivors' Testimonies project; members of the "Learning how to Learn" team; and Susanne FRIESE for her critical and constructive reading of an early version of this paper.

References

African Rights (1995). *Rwanda: death, despair and defiance.* London: African Rights.

Appelt, Wolfgang (2001). What Groupware Functionality Do Users Really Use? Analysis of the Usage of the BSCW System. In *Proceedings of the 9th Euromicro Workshop* on PDP 2001, Mantua, February 7-9, 2001. IEEE Computer Society, Los Alamitos. Available at: http://bscw.gmd.de/Papers/PDP2001/PDP2001.pdf.

Bentley, Richard; Appelt, Wolfgang; Busbach. Uwe; Hinrichs, Elke; Kerr, David; Sikkel, Klaas; Trevor, Jonathan & Woetzel, Gerd (1997). Basic Support for Cooperative Work on the World Wide Web. *International Journal of Human-Computer Studies*, *46*, Special issue on Innovative Applications of the World Wide Web, 827-846.

Baeza-Yates, Ricardo & Ribiero-Neto, Berthier (1999). *Modern Information Retrieval* New York: ACM Press.

Black, Paul & Wiliam, Dylan (1998). *Inside the Black Box: Raising Standards Through Classroom Assessment*. Available at: http://www.kcl.ac.uk/depsta/education/publications/blackbox.html.

Bowen, Kenneth (1991). *Prolog and Expert Systems* New York: McGraw-Hill.

Bray, Tim; Hollander, Dave & Layman, Andrew (1999). *Namespaces in XML*.Available at: http://www.w3.org/TR/REC-xml-names/.

Brian, Dan (2000). Lingua:Wordnet. *The Perl Journal*, *5*(2), 40-48.

Christiansen, Tom & Torkington, Nathan (1998). *The Perl Cookbook*. Sebastopol, CA: O'Reilly and Associates.

Cordingley, Elizabeth (1991). The upside and downside of hypertext tools: the KANT example. In Fielding, Nigel G. & Lee, Raymond M. (Eds.), *Using Computers in Qualitative Research* (pp.pp.164-180). London: Sage.

Cross, David (2001). *Data Munging with Perl*. Greenwich. CT: Manning.

Fellbaum, Christiane (Ed.) (1998). *Wordnet: An Electronic Lexical Database*. Boston, MIT Press.

Fielding, Nigel G. & Lee, Raymond M. (Eds.) (1998). *Computer Analysis and Qualitative Research*. Thousand Oaks, CA: Sage.

Fischer, Michael (1994). *Applications in Computing for Social Anthropologists*. London: Routledge.

Floyd, Michael (1999). *Building Websites with XML*. Upper Saddle River, NJ: Prentice Hall PTR.

Francis, Brian; Homer, Alex & Ullman, Chris (1999). *IE5 Dynamic HTML: Programmer's Reference*. Chicago, IL, Wrox Press.

Friedl, Jeffrey (1997). *Mastering Regular Expressions*. Sebastopol, CA: O'Reilly and Associates.

Glaser, Barney (1978). *Theoretical Sensitivity*. Mill Valley, CA: Sociology Press.

Goldfarb, Charles (2000). *XML in an Instant: a non-geeky introduction.* Available at: http://wwwxmlbookscom/press/nongeekyhtm [Broken link; September 2002, FQS].

Gourevitch, Philip (1998). *We Wish to Inform You that Tomorrow we will be Killed with our Families*. London, Macmillan/Picador.

Human Rights Watch (1996). *Shattered Lives: sexual violence during the Rwandan genocide and its aftermath*. New York: Human Rights Watch.

Keane, Fergal (1996). *Season of Blood: A Rwandan journey*. Harmondsworth: Penguin.

Macherius, Inigo; Rodrigues, Michel; Derksen, Enno; Almeida, Jose & Josten, Geert (2000). Ways to Rome: Processing XML with Perl. *Proceedings of YAPC 19100, Carnegie Mellon University, Pittsburgh PA, 26-28 June 2000,* 286-293.

Marchal, Benoit (1999). *XML by Example*. Indianapolis: Que.

Melvern, Linda (2000). *A People Betrayed: the role of the West in Rwanda's genocide*. London: Zed Books.

Muhr, Thomas (2000, December). Increasing the Reusability of Qualitative Data with XML [64 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* [Online Journal], *1*(3), Art. 20. Available at: http://www.qualitative-research.net/fqs-texte/3-00/3-00muhr-e.htm.

Prunier, Gerard (1995). *The Rwanda Crisis 1959-1994: history of a genocid.* London: Charles Hurst and Co.

Quin, Liam (2000). *Open Source XML Database Toolkit*. New York: John Wiley and Sons.

Richards, Thomas & Richards, Lyn (1994). Using computers in qualitative analysis. In Norman K. Denzin & Yvonna S. Lincoln (Eds.). *Handbook of Qualitative Research* (pp.445-462). Thousand Oaks, CA: Sage.

Rodrigues, Michel (1999). A review of Perl-XML modules. *Proceedings of YAPC 19100, Carnegie Mellon University, Pittsburgh PA, 26-28 June 2000*, 294-311.

Thoeny, Peter (2000). Corporate Collaboration with Twiki. *Web Techniques, 5*(12), 51-55.

Udell, Jonathan (1999). *Practical Internet Groupware*. Sebastopol, CA: O'Reilly and Associates.

United States Holocaust Memorial Museum (1998). *Oral History Interview Guidelines*. Washington DC, United States Holocaust Memorial Museum.

Wall, Larry; Christiansen, Tom & Schwartz, Randal (1996, 2nd edition). *Programming Perl.* Sebastopol, CA: O'Reilly and Associates.

Weitzman, Eben & Miles, Matthew (1995). *Computer Programs for Qualitative Data Analysis*. Thousand Oaks, CA: Sage.

Author

Dr *Patrick CARMICHAEL* is a lecturer in the School of Education at the University of Reading, UK, where he develops and evaluates network technologies for use in education and other civil society projects He provides IT support to *Survivors' Fund*, a UK-based NGO supporting survivors of genocide in Rwanda, and has contributed a chapter, *Information Interventions, Media Development and the Internet*, to "Forging Peace: Information Intervention, Media and Conflict" edited by Monroe Price and Mark Thompson (EUP, 2002). He is also a member of "Learning how to Learn", a four-year Economic and Social Research Council (UK) Project which involves research into the representation of practitioner knowledge across electronic networks.

Contact:

Dr Patrick Carmichael

School of Education
University of Reading
Bulmershe Court
Woodlands Avenue
Reading
RG6 1HY, UK

E-mail: p.carmichael@reading.ac.uk

Citation

Carmichael, Patrick (2002). Extensible Markup Language and Qualitative Data Analysis [39 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *3*(2), Art. 13, http://nbn-resolving.de/urn:nbn:de:0114-fqs0202134.

Revised 2/2007