The C-ODA Project: Experiences and Tools

PETER KIRSTEIN AND GOLI MONTASSER-KOHSARI

Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK.
Email:kirstein@cs.ucl.ac.uk,gmontass.@cs.ucl.ac.uk

In this paper, we discuss the UCL the C-ODA project, working with a large database of journal articles of chemical journal in several compound document forms (text/image). As part of the project, there is a need to convert a large mass (approximately 500 000 pages of technical papers) of SGML documents into an ODA representation; our tools for, and experiences in, converting these articles are described. We provide a number of interfaces to access that data, including WAIS, PixLook and SuperBook. Access is provided over various forms of network—LAN, Internet and ISDN in particular. Our experiences with putting up the databases, and with the users accessing the data is presented.

Received February 3 1995, revised June 19 1995

1. BACKGROUND AND OVERVIEW

1.1. Overview of the project

The American Chemical Society (ACS), Bellcore, Chemical Abstracts Service (CAS), Cornell University and OCLC are collaborating in the CORE project [1] to deliver electronic information from primary publications to end-user chemists.

As part of this experiment Bellcore have scanned approximately 500 000 pages of ACS journals published between 1982 and now. They have also processed the database tapes derived from the typesetting process for the same journal issues into a Standardized General Mark-up Language (SGML) [2] format so that it may be indexed and/or typeset. They have provided a large electronic database containing approximately 100 000 articles, representing 500 000 pages of journal articles of the American Chemical Society (ACS). The data is held at the Cornell University Mann Library for access over local area networks (LANs) by Cornell chemists.

In the Computer Science Department of University College London (UCL-CS), we have been involved with the CORE project since 1988. This activity relies heavily on the work of Bellcore, and uses the data provided by the ACS. It is supported both by British Library Research and Development Department (BLRDD). While we have provided facilities similar to the CORE project, we have also been interested in areas on which the CORE project has not focused. The UCL activity is referred to as the C-ODA project, and covers also areas such as applications of ISO standards, and usage of relatively low-bandwidth networks such as the ISDN. In the C-ODA project, we are providing access to the ACS material, and also to a few issues of an electronic journal of the British Institute of Physics, *Modelling and Simulation in Materials Science and Engineering* (*MSMSE*). This paper discusses the way the database is set up—which involves conversion from a SGML representation into open document architecture (ODA) [3], the methods of indexing, the access methods provided, and our user experience. We also discuss the reasons for many of our implementation choices.

The ACS has consented to allow the data to be used for these projects, with certain restrictions on distribution—mainly that the data will not be available outside Cornell University for the CORE project, and outside the University of London for C-ODA.

This project started in 1991, when UCL-CS was heavily involved with ESPRIT PODA projects (e.g. [4], [5]) in the use of ODA. At that time, the CORE project was using no standard language for the representation of the text, so that ODA was a natural choice for the C-ODA project. Later the ACS textual material became available in SGML form. Even then, there are significant advantages in the use of ODA, which are discussed in Section 2; for example, ODA is a blind open interchange format for which a number of converters are available, unlike SGML, in which the interchange is dependent on the DTD. Our wish to continue with the ODA formulation gave us the need for the SGML → ODA convertor. On the whole, the ACS SGML documents share the same DTD, and so we had a range of options for writing our SGML to ODA converter.

Having an on-line database of scientific journals offers many advantages over conventional paper-based journals; many of these advantages fall into the areas of search and access. Electronic searching texts for information is much easier than manual; far more productive searching can be undertaken using a computer system. In our environment all the journals are indexed so that, despite the size of the database, searches are very fast. Electronic access provides additional advantages:

- It is non-exclusive—any number of people can access the same journal simultaneously.
- It is distributed, so it is not necessary to be in close proximity to the database in order to access its information.
- It can be integrated with the users' facilities, so that it is possible to extract information for other

purposes—always subject, of course, to consideration of copyright and other constraints.

We have set up a document database which can be queried in a convenient manner, and allows the user to browse the results on screen using a number of different tools. We have provided facilities for end-user chemists to access the database at various locations within UCL. A portion of the data was provided originally in the same form as in the CORE project; now, the database is supplemented by transforming all the data which we have into the ODA/ODIF format, and making it available to the University of London (UL) chemists in that form. At present we are using a large set of the 1982–94 collection of ACS journals, providing a number of interfaces to access that data, including several using WAIS [6] (a system based on ANSI Z39.50[7] information storage and retrieval protocol), and a tool developed at Bellcore called PixLook [8]. We are additionally evaluating the use of a Hypertext Browser called SuperBook [9], also by Bellcore. We will be using a substantially larger data set when we have all the data available.

When we started the C-ODA project, due to the size of the dataset, the most sensible device for storing the documents was an Optical Juke Box (JB); hence the department acquired a 90GB HP magneto-optical JB for this purpose. With the more rapid reduction of the cost of magnetic storage than magneto-optical, this may no longer be the case. We have developed a JB interface library which virtualizes the JB as a single large storage device, so that the application programs do not need to track the locations of files among the discs in the JB, to which a high speed storage server, with some 18 GB of disc space is attached as front end. A reverse index of all the document text is held in the disc storage. For the whole 10 years of data this will contain about 4 GB (see Section 3.2). All searching of document contents is done from the disc storage; the retrieval of the documents themselves is from the JB which holds the documents in all forms. Finally, to assuage the worries of publishers we felt it essential to add various forms of integrity control, authentication and audit trails; this activity is not yet complete (see Section 6).

1.2. CORE and C-ODA differences

The initial UCL activity was funded jointly by the CEC under the ESPRIT PODA-SAX project, and by the BLRDD under the C-ODA project. PODA-SAX was concerned with piloting the use of ODA, and C-ODA contributed the largest and most complex database of ODA documents yet held.

The C-ODA project mirrors that in the CORE project in the use of the same database on the UL sites. It has, however, some significant differences.

C-ODA uses the ODA form to represent all the data; CORE uses SGML for the text, and its own format for the images. Moreover, most of the time the text representation is used only for the document search; the data access is normally to the bitmap, page-image forms of the whole data. The CORE methods use a data representation closer to the original journal, but much more voluminous.

CORE uses exclusively LAN access at 10 Mbps to the database; remote access to the database is not very satisfactory due to the concentration on bit-map form of data, with its consequent size. C-ODA concentrates on the compound document form (in ODA), and so is able to provide access both over the London University wide area network (WAN), which is part of the Internet with lower access speeds, and over the ISDN at 64 Kbps.

CORE uses a Sony JB with 3 GB write once read many (WORM) 12′ platters (1.5 GB per side); C-ODA uses a Hewlett Packard JB with 600 MB write read many times 5′ platters. The UCL JB based on the smaller platters has been more reliable; moreover, because the Cornell U JB is WORM, they have had to be much more careful about what they write on the platters than UCL-CS.

CORE uses the OCLC Newton search engine to locate articles; C-ODA uses the WAIS text search engine. The feel of the Newton search engine is more familiar to conventional retrieval librarians in its use of field searching than the public domain (PD) version of WAIS, and is capable of dealing with larger databases; the commercial version of WAIS has similar facilities to Newton, and has no problem with the large databases with which the PD version has difficulties.

CORE has access to the ACS Chemical Abstracts Service (CAS) data, not initially available to C-ODA; the chemists like access to these data.

1.3. Activities in the C-ODA project

The CORE project is primarily concerned with high-bandwidth (>10 Mbps) access to this dataset, and so is concentrating its efforts upon full-text retrieval of scanned images. The C-ODA project is interested in extending this work to lower-bandwidth communications like the 64 Kbps ISDN-2 system; the ODA encoding of the documents offers many advantages for this approach.

The C-ODA project had two main strands: replicating the work undertaken by Bellcore and its partners in the USA, and also extending the work into the new direction of the ISO ODA open document architecture, and taking advantages of the flexibility that this route affords. The starting point for both strands is the work of Michael Lesk at Bellcore who has built a number of tools to convert the original ACS data from the database tapes derived from the typesetting process, which is in a proprietary form, into a standard form used by publishers—the SGML format [2]. He has augmented the text with scanned images of the journals and diagrams [10] to form the ingredients for a rich text-image database.

We provide a document database which can be

queried in a convenient manner, and allows the user to browse their results on screen using a number of different tools. We have provided facilities for end-user chemists to access the database at various locations within the university. Originally the data were provided only in the same form as in the CORE project; but now, the database is supplemented by transforming all the data which we have into the ODA/ODIF format, and making it available to the chemists in that form. The interfaces provided to access that data include WAIS, SuperBook and PixLook. We are also evaluating how SuperBook can be extended to give intelligent Hypertext guidance to users [11].

The work we have undertaken in this project is as follows:

- Develop a flexible converter from SGML into ODA that can be used with any DTD (see Section 2).
- Due to its size, the image data is stored on an optical jukebox (JB); we have developed a JB interface library which allows the JB to be considered as a single large storage device (Section 3).
- Replicate the Bellcore/OCLC work at UCL, and extend the interface tools to use the ODA representation. The Bellcore tools do not adequately deal with the problem of text and graphics on the same page, whereas the ODA-based viewers provide a much more natural presentation of such material. (Section 4).
- Provide remote access to the database over basic rate ISDN.

1.4. Overview of the publishing chain

While work with the ACS databases as processed by Bellcore were the main activity in the project, we obtained a good insight into how the publishing chain should proceed for this type of activity. The fact that it did not always do so only made our task harder.

The conventional publishing chain for journals in science and engineering is as follows. Journals are submitted in a number of forms by the authors. The chosen format by authors seems to be predominantly TeX or LaTeX and Postscript, but this is not always the case. The articles can be translated into a proprietary mark-up language (with a specific DTD) for typesetting, and then printed. The way ACS produced its journals up to the end of 1994, was that the diagrams were stuck on to the masters before printing. This meant that the database tapes derived from the typesetting process did not include the diagrams, although they did include equations and tables. For a full electronic form, the figures must also be provided electronically.

A distribution format should have the following properties as a minimum:

- *Presentation*: it should contain presentation information sufficient to generate a pleasing image for the reader. For example, it should enable titles and headers to be in larger font, and allow for typographical effects such as italicizing and boldness.
- *Content*: it should contain the words of the article (or possible the front matter of the article) in order to facilitate searching.
- *Viewing*: tools should be available for readers to view the system on screen, and possibly generate hard copy as well. These tools must be friendly, reliable and well supported.

Electronic journal (EJ) delivery involves a publisher generating documents and distributing the electronic form to organizations which will pass these on to the users. For the sake of argument we will call these organizations 'electronic libraries', even though they may not be what are currently recognized as libraries.

For an electronic publishing chain, instead of being printed, the data are converted into a form which is suitable as a distribution format, and then sent to the 'electronic library' organizations. The reader of these documents will require them in one of two ways. Either they will be receiving a new issue of the EJ, in which case they will wish to inspect the table of contents, browse the articles, and/or read a number of articles in depth. Alternatively, they will wish to search against a collection of journals, using some kind of query mechanism, and then browse or read the articles that were found. However, it is also possible that a reader may wish to browse old journals, or search in a new issue, and the user should be able to do both.

When viewing the EJ, the reader will expect the articles to be clear and contain formatting suitable for supporting the document structure. Moreover, all readers and screens are not equal, and so some method of changing the size of the documents would be advantageous.

1.5. The source data

Since 1977 the American Chemistry Society has preserved an increasing proportion of the tapes used to typeset its journals; most of them have been preserved since 1982, though the format has changed slightly over the years. These tapes contain all the textual information of the journals, including highlighting, equations and tables, and also a large amount of contextual information. This contextual information includes what we describe as document management attributes (DMAs), and also some of the structural information of the articles. The current tapes used for the typesetting process do not, however, contain any of the graphical images, or any layout or presentation information. Bellcore have obtained the graphic images by scanning the microfilm copies of the published journals and using custom OCR techniques to identify page components such as figures, tables and schemas (captionless figures) since no other record of the images is available.

Until 1994, the format of the database used for the typesetting process has been a proprietary scheme encoded in an IBM database format. This is converted into SGML by Bellcore as part of the CORE project. They pass the SGML versions of the documents on to us

(along with the scanned image components), with the permission of the ACS. This is in a special DTD used only for this data, but based on the American Association of Publishers AAP DTD. We gratefully acknowledge this assistance from Bellcore and the American Chemical Society.

The tables and equations are not translated from the database tapes used for the . Instead, the graphics, tables and equations are derived from the scanned page images in bitmap form. When this process had been completed, there are two data sets, one representing the text in SGML, the other figures, tables and equations. The extracted graphics activity was quite error prone; a 95% success rate at finding figures has been achieved in the past, but this is being improved.

2. SGML ODA CONVERTER

2.1. Comparison of SGML and ODA

SGML is a system of specifying generic mark up for documents. The point of generic mark up is that it denotes what an element represents, rather than what it looks like. The mark up should describe a document's structure and other attributes rather than specify the processing that is to be performed on it, as descriptive mark up need be done only once and will suffice for all future processing. For example, one would mark the title of an article with the tag <title>, rather than saying 'centred, bold, 16pt Times Roman'. The description of <title> is then contained in a document type definition or DTD.

ODA supports this functionality using a mechanism called a document class but also allows presentation information to be bound to the document elements. ODA has been designed primarily as an interchange format for documents. ODA is well supported by commercial wordprocessor manufacturers, and converters are available between ODA and commercial wordprocessor formats.

SGML uses an ASCII-based representation which has certain in-built limitations. In particular, it is not possible to embed arbitrary binary data within an SGML document, since elements are terminated by a special character sequence—and clearly that sequence is possible in arbitrary binary data. It is possible to circumvent this using escape sequences, but there is no defined way to do this within the ISO SGML standard. The accepted method is to refer to external entities for such items. ODA uses a binary representation expressed as ASN.1 streams; as such it is not subject to such restrictions. It is, however, interesting to note that one of the reasons why SGML is well used is because it is easy to generate the ASCII representation; on receiving an SGML file, it is possible to scrutinize it effectively using just a standard text editor. It is usual to refer to an ODA encoded in this way as an ODIF (open document interchange format) file.

ODA is a more suitable format for document distribution than SGML for the following reasons:

- A single ODA file can encapsulate a compound document; its distribution as ODA only requires a single file to be passed, whereas a compound document in SGML is likely to consist of a number of separate files.
- The ODA file contains enough information to render the file on screen or paper in a pleasing and meaningful manner. SGML requires that the DTD and a translation specification file be sent.
- The viewing tools for both ODA and SGML data are of similar quality. However, the SGML viewing tools have different types of translation specification file; such a file would be needed for each viewing tool which end-users intended to use. The take-up of DSSSL [11] will remove this difficulty, but for the next year or two, this will be the problem.
- ODA can be readily converted into a wide range of commonly used wordprocessing formats. For example, there are converters available which convert ODA in WordPerfect, Microsoft Word, Microsoft Word for Windows, IBM DisplayWrite, DCA-RFT and DecWrite formats. It is possible for a system which holds documents in ODA to deliver them to users in a format which they can view on their normal equipment. Moreover, they can edit these documents annotate them, or extract parts into their own documents all within their normal document processing environment.
- The ODA format is reasonably compact. The format supports geometric graphics, and bitmaps are compressed using the Group 4 fax algorithm—an excellent lossless compression scheme—or Group 3 fax algorithm or bitmap. The SGML equivalents are stored in TIFF; in our case this has been transformed first to bitmap and then compressed using type 3 facsimile compression.
- The ODA format does not suffer the ASCII-related problems with which SGML files must contend. The ODA files do not need altering when files are transferred between ASCII and EBCDIC-based machines, or between machines with different byte orders, or between ASCII-based machines with different line break characters (for example between DOS and UNIX).
- There are limits in the flexibility of changing font sizes in ODA; but fonts are not supported in the SGML format itself.

Many of these consideration are not applicable to the initial generators of the SGML document; SGML is an excellent authoring format, due to its more sophisticated data-modelling potential. We have found that the concept of authoring in SGML and distribution in ODA brings together the best of both worlds.

2.2. The SGML to ODA conversion

With limited resources we would not have been in a position to develop the ultimate SGML to ODA

converter, and we would have been foolish to attempt this. A number of previous reports on SGML and ODA interworking have clearly indicated that imposing a few constraints greatly simplifies this task. We chose to focus on a one-way conversion from SGML to ODA, and to largely disregard providing any support for the resultant ODA to be converted back into a similar SGML. We also chose to largely disregard maintaining the structure of the document; we flatten the document hierarchies. The emphasis of the converter is on the presentation aspects; our aim was to produce an ODA document with presentation attributes that look correct.

The ODA document will not normally look identical to its SGML counterpart, since SGML does not contain any presentation or layout information. In order to convert the SGML to ODA we need additional information which will specify how elements are to be presented, and what other action are required by elements. This is a standard concept in SGML publishing chains. An SGML DTD, a document instance of that DTD, and a translation specification are the inputs needed. The SGML instance document is validated against the DTD, then the translation specification is applied to the SGML in order to produce an output which is the SGML content with the layout and presentation applied. Thus SGML is published by applying layout and presentation rules for the elements to the SGML. The resulting information is represented in ODIF. ODIF is a very complex standard, and a number of restricted levels of functionality, called document profiles have been defined in the PODA and previous projects. While we used other profiles earlier, we now generate ODIF according to FOD26 [12], which has much better fonts supports than earlier versions.

The following sections define the styles and effects used to specify how elements can be presented, and then a third section describes the mapping rules which are used to bind elements to these presentation styles, and direct the flow of the text onto the resulting ODA document.

2.2.1. *Style specifications*

The style specification consists of a number of declarations which define either styles, or effects. Styles are lists of attributes which can apply to a section of text in a document. A given style defines all attributes and therefore any two pieces of text with the same style have equivalent presentation attributes. When styles are defined, some attributes can be inherited from another style (no more than one). If a style is defined which has no name, then that is treated as the base style; if a style is not explicitly based upon another style, then the base style is used as the style to inherit attributes from. An example of styles is given in Annex 1.

2.2.2. *Effect specifications*

Effects are like styles, except they do not define all attributes. When an effect is invoked, then the undefined values are inherited from the currently active effect, i.e. only the attributes specified in the effect are changed. When defining effects, effects can inherit attributes from any number of other effects. If two parent effects both define the same attribute, then the definition given in the latter is used. Any number of effects may be applied to a style. There is no base effect. Example effect definitions are as follows:

```
EFFECT bold
{  FACE = Bold;
}
EFFECT italic
{  FACE = Italic;
}
EFFECT bolditalic : bold : italic
{No content just use what the parents have
}
```

The full set of presentation attributes are available elsewhere [13].

2.2.3. *The translation specification*

The translation specification maps rules to SGML element (tag) names, and allows certain contextual information to trigger which rule is to be used. The syntax and semantics of the translation specification language are influenced by DSSSL, but it has significant differences from that and most other SGML layout systems. Primarily, the contextual information used to select rules is the name (tag) of the element, the parents in the SGML element structure, and any attributes of the element. For example, one could define a rule which was only triggered for elements tagged with <highlight> which only activated inside an <abstract> element. Alternatively, one could define a rule which only activated when, say, the level attribute was set to 1.

A mapping rule consists of a number of Directors. Directors are sequences of actions which are applied to an element. There are three types of director; start directors, usage directors, and end directors. These are activated when an SGML element (tag) opens, contains data, and ends respectively. Each director outputs to a specified receiver. The most common example of a receiver is the main text TXT. Other receivers are headers HDR, footers FTR and document management attributes DMA:name. The following example gives a feel of the structure of a typical mapping rule:

```
MAP titleDashDash indicates comment to
  end-of-line
{ U(STYLE title) > TXTForce to use title
  style
U() > HDRUse default style for this object
U() > DMA:TITLEAlso place this data in the
  Title DMA
E(NEWPARA) > TXTThrow new paragraph
E(CLOSE) > HDRClose HDR (so cannot be
  altered)
```

```
E(CLOSE) > DMA:TITLEClose Title DMA
  (so cannot be altered
)
}
```

The items within the curly braces {} are the directors. They consist of the letter S, U, or E, followed by a number of actions in the round brackets, and optionally followed with a greater than sign '>' followed by the name of a receiver. Note that the same data can be sent differently to each receiver. The map rules can multiplex the input so that it appears in several places in the ODA document—as is useful for a <title> element above. There are a wide range of actions, which for example apply paragraph styles, character styles, insert additional data into the output, and insert line or paragraph breaks. The styles and effects are defined elsewhere in the same file.

The approach described above has worked very well for simple documents, and handles most aspects of the ACS SGML data very well. Its main omissions are embedding external graphics images, and rearranging sections of the output. We support rearrangement by adding a new receiver type, called a store (STO:name) which accumulates output from directors. There are also a pair of new actions called recall and recall-deferred which insert stored output at the current point, and at the next paragraph break respectively. Nesting is not allowed currently, but later implementations will allow this.

2.2.4. Implementation

Writing an SGML parser is a very difficult task, and we were pleased that we could utilize the work of others for this part of the converter. Goldfarb, the primary force behind SGML, has released a public domain SGML parser called arcsgml; this has been improved upon by Clark into a new tool called SGMLS [14]. While SGMLS is still under development, version 1 is stable in the functionality it provides, and is a usable base for this project. Essentially, SGMLS reads a DTD and an SGML document, validates the document against its DTD, and generates an ESIS describing the document. The ESIS generated is a linear ASCII data stream with records separated by newline characters. The ESIS is very easy to parse, and requires no validation. Each line begins with a distinguishing character describing the data which is to follow. For example an '(' indicates that an element is opening and is followed by the generic Identifier (GI) of the element; a ')' indicates that an element is closing (again followed by the GI), a '-' indicates that the following line is document text (i.e. not mark up), and an 'A' indicates an attribute of an element that has been set, or inferred.

Similarly, it is not trivial to write a system which generates valid ODIF, and we were fortunate to be able to build on an existing system at UCL which converts between ODIF and files from the BBN Slate Multimedia Document Editor [15]. The back-end of the Slate to ODIF converter was reused for the SGML to ODA converter. This code also requires the ISODE toolkit [16] in order to function.

The new code goes in between these two existing elements. It performs the following tasks:

- Reads and validates the translation specification file.
- Reads the ESIS from SGMLS, and records the current context in the document hierarchy as elements are opened and closed. If we think of the document having a tree structure, then this context is the path from the current position in the document to the root of the document, along with the attributes associated with each element on that path.
- Applies mapping rules to elements when the relevant element occurs in the ESIS. Invocation of a mapping rule may create a new 'receiver'.
- The content of an element must be dealt with according to the rules currently in force for each open receiver.
- When an element closes, the rules which applied to the parent element are in force once more. Also, some receivers may have now gone out of scope, and as such, they must be de-allocated.

3. STORING DATA

3.1. The use of an optical jukebox

We have installed a large document store, consisting of a Hewlett Packard optical JB with 4 Sony drives, a Sun SparcStation (Sparc-5 with 96 Mbyte of primary store) as a dedicated server, and 18 GB of magnetic storage. The main storage consists of 144 magnetic optical platters each with 600 Mbyte of data; this allows 90 GB of rewritable storage. Access to arbitrary data is slow—15 s. However it is possible to stage the data into the disc storage.

At the moment we are managing the data on the JB ourselves. Some of the more recent JB software allows an application running on a Sun Sparcstation to access transparently any disc in an optical JB via standard Unix functions. It treats the whole JB as an integrated disc store—while still giving us some control on what to cache in the magnetic store. We are still investigating the advantages of that type of software.

We store all the text data on the front-end magnetic storage. This allows content searching to be done relatively fast. The full ten years of data, will require approximately 3GB of storage.

It is an important aspect of the C-ODA project that the JB uses magneto-optical rewriteable storage. The CORE project uses write once read many (WORM) storage; as a result, CORE is concerned about getting the data right before it is put onto the JB. Since we have found that it requires many passes through the whole data in practice, this has had the impact of making all the data manipulation a very long-winded process; CORE

has usually worked for a longer time with smaller databases on disc store, and been very hesitant to commit to using the JB.

3.2. Database sizes and access times

We now have considerable experience on the size of the data, and on the access times. We have the text component of the database for most if 1982–94, and the bitmap form for much of 1988–94. The exact data now up is given below:

TABLE 1. Number of articles on line

Years	*No of text articles*	*No of bitmap articles*
1982–88	4846	0
1989	9826	1097
1990	11392	2000
1991	15825	7176
1992	15725	8000
1993	15297	5889
1994	1559	976

We are expecting shortly more SGML and the extracted images for 1994, and have some 50 GB of page image data for 1991–94 which we are in the process of loading onto the JB. From the above it is clear that the actual data management of these large collections, when they pass through so many stages of processing, is difficult.

Working with the whole database of 1980–94, we have 4 GB of SGML, 5 GB of ODIF, and 1.3 GB of extracted figures. We treat each period mentioned in Table 1 as a separate database, and the search for any particular word combination is done on each database. Thus, for example, searching for any single word (e.g. Robb), would take less than a second on each database; in one such search, 847 documents were found. It is also possible to do a field search on the same data; if the same database was searched in a field sense (e.g. author = Robb), then the search time was little changed, but the number of documents retrieved was more manageable and *precise*—only 23 documents.

A typical comparison of the data sizes and access times of typical articles in the SGML and ODA formats, and the figures sizes are given in Table 2.

Here the SGML and ODA give the sizes of the stored text, while the figures size gives the compressed stored image. The display size shows the data which has to be transmitted. The access times include retrieval; the conversion time include decompression on a Sparc-5 WS.

TABLE 2. Sizes/access times of typical articles

SGML KB	*Figures KB (nos)*	*ODA KB*	*Display KB*	*Access sec*	*Conv sec*
67.8	xxx	66.2	74.8	3	2
14.2	2.1 (1)	19.5	45.7	2	2
59.9	46.2 (8)	110.7	424.9	2	3

4. USER INTERFACES

4.1. Introduction

Having an on-line database of scientific journals offers many advantages over the conventional paper-based journals; and many of these advantages fall into the areas of search and access. Much of the UCL-CS interest in the project is in providing different means of search and access, and gauging the comparative value of the different methods.

Electronic searching texts for information is much easier than manual; far more productive searching can be undertaken using a computer system. In our environment all the journals will be indexed so that, despite the size of the database, searches will be very fast. Most of the user interfaces we offer will support full-text retrieval—every single word in the document is indexed so that the searches go beyond any keywords that the author/classifier has deemed appropriate. Again, search responses are virtually instantaneous.

Electronic access provides additional advantages. It is non-exclusive—any number of people can access the same journal simultaneously. Access is distributed—it is not necessary to be in close proximity to the database in order to access its information. Access can be integrated with the users' facilities—it is possible to extract information for other purposes.

Most search requests are based on some type of word-based search, the system looking for occurrences of the words in its document base. Searches may be restricted to certain kinds of data in the documents such as titles, author names, or abstracts—or may be applied to the whole of the text in the document. One of the interfaces (WAIS) support relevance feedback—this allows the user to mark one or more documents in the database as being relevant to the query and the search algorithms will favour similar/related documents subsequently.

Algebraic text searching allows greater control over text queries. Algebraic text searching allows the user to specify rules about how the documents are to be searched. Say a search is looking for the words 'petroleum' and 'refinement'. The number of documents containing both words could be quite high, although there is no guarantee that a document containing both words may be about the refinement of petroleum—the occurrences could have been on separate pages. However, if the search were to look for 'petroleum' and 'refinement' in the same paragraph, then one would expect a higher 'hit rate' of appropriate documents. Some of the interfaces allow algebraic searching.

One of the key differences between the work being done in the Cornell University CORE projects the USA and the UCL centre is the network access. The CORE project is concentrating upon high-bandwidth LANs, which can deliver large amounts of data rapidly; hence they emphasize the bitmap representation of the journals. At UCL-CS we are particularly interested in widening the scope of the project to include remote

access to the document database often involving relatively low-bandwidth communications e.g. basic rate ISDN lines at 64 Kbps. At this speed a typical page in bitmap form, occupying 100 KB, takes at least 12 s to deliver. However delivery of the document form would be nearer 1 second per page, or perhaps three or four seconds if images were also transmitted.

Because we want to provide the technology to make access to this database possible outside the high-bandwidth local area network at UCL—even if the ACS constraints do not allow us to offer such a service outside the University of London. This remote access gives a strong emphasis upon the document form of the journals. Bitmap delivery is also possible, although it is slower and hence less convenient in these circumstances. We expect to introduce at a later stage other document collections, which have less constraints on their usage than the current ACS ones.

4.2. The user interfaces

We have been offering a number of user interfaces to the journal database, many coming from the CORE project. The following paragraphs describe each of these alternatives. Those which require X windows can either be run on a UNIX workstation, an X-terminal, or a PC with X-terminal capability.

4.2.1. PixLook

PixLook [8] is a purpose-built tool written by Mike Lesk for the ACS project. It allows the user to specify simply a number of keywords and then looks in its index for documents associated with those keywords. It then presents a bitmap image of that page, and allows easy key-presses to move around the page, Zoom in/out, and move forward and back pages. PixLook works under the X Window system. PixLook will only work on a local-area network (it needs direct access to the journal files).

4.2.2. WAIS and Xwaisq

WAIS is the wide area information server tool developed and placed in the public domain by Thinking Machines Corp [6], and now being developed further as a commercial product by WAIS Inc. WAIS provides tools for full-text indexing different types of data, and allowing that index to be queried by a remote machine. It is a classic client–server system with a back-end (the WAIS server) which searches an index based upon queries provided by a front end (WAISQ—WAIS question). The WAIS server can provide both lists of documents with their 'scores' according to some query, and whole documents when a user selects a document from a list. Xwaisq is an X-based question program which is provided with the WAIS distribution.

The WAIS programs have been extended to display both bitmap and document representations of the ACS journals and the postscript and encapsulated postscript of IOPP.

4.2.3. SuperBook

SuperBook [9] is a general purpose Hypertext tool developed by Bellcore which has particularly strong support for information with a mainly hierarchical structure. Lesk has developed tools to convert from the SGML format into the mkbook' format which is used to generate SuperBook databases. SuperBook works under the X Window system. SuperBook is a client–server tool and so the client does not require the filestore containing the data to be locally mounted and therefore it can be used remotely.

5. USER EXPERIENCE

The user experience is still limited. Feedback is on a casual basis, either in person, or via email or by telephone. The CORE project is doing more formal user tests. The following highlight the immediate concerns of the users after a few half hour sessions with tools.

- Immediacy of access is more important than quality of access. Although chemists are prepared to travel to the Computer Science Department in order to take advantage of the workstation screens, they really like more immediate access to the data on cheaper workstations via lower-bandwidth lines.
- A critical difference from paper-based systems is the ability to automatic follow references. An elementary scheme in PixLook whereby you press a key to get the list of references to and from the current document, and then click on one to view, is particularly desirable. Particularly if an article has a later correction to it, then automatic linking/referencing to the correction is extremely useful.
- The ability to scroll a highlight through a search list is important because this automatically tracks the place in a list of documents. When shown the Xwaisq selection window, users identified that this was more appropriate.
- The need to view, edit and augment previous searches was considered to be very important. The lack of such a feature seems to discourage casual browsing.
- The lack of scroll bars on the right hand side of windows was considered an important omission.
- Some of the chemists who are familiar with on-line databases are keen to use the registry numbers provided by the ACS.
- Users like the 100 dpi size for browsing, but considered it inappropriate for reading. Similarly speed is considered good for these images. However, when shown the regenerated text from the database tapes (SuperBook and Xwaisq:read), they thought this was a major improvement.
- Paper is still considered to be the best form for reading

a journal article in depth. Users did not feel that they would be happy to absorb a journal from the screen.
- Some of the pages even at 300 dpi have unusable pictures.

6. SECURITY FEATURES ON DATABASES

Secured telematic documents are relevant to protecting both prosecutable and formatted documents when transmitted for continued processing by the recipient using computing equipment, rather than fax. The security extensions communicated enclose the conventional telematic document in a protective seal, processes in the local work station, combined with key distribution service, will release contents only to authorized recipients.

The OSISEC [17] is a security package developed at UCL which implements the services described in the X.509 Authentication Framework. These comprise data confidentiality, data integrity, origin authenticity and non-repudiation of data origin.

One of applications of OSISEC is a package called DOCSEC [18], which provides the following security: services to the documents:

- *Confidentiality*: ensuring that the content of a document or part of the document is only disclosed to specified recipients.
- *Integrity*: ensuring the privileged recipient that a given document or part of the documents has not been tampered with.
- *Authenticity and non-repudiation of origin*: proving that the originator is the source of a given document or part of document.

Confidentiality on a document in the database will not serve any purpose. Integrity on documents in the database provides the recipient with a way of ensuring that the documents are integral, i.e. it assures the recipient that a given document has not been modified by someone unable to provide the integrity check. Authenticity on documents in the database establishes that the claimed originator is the source of a given document, although it is also possible to create a database in which parts of the documents are made secure to unauthorized recipients.

7. CONCLUSIONS

In these conclusions we use the term 'small' for a database of 284 documents, 'medium' for one of 10 000 documents, and large for one of 85 000 documents—the whole ACS database since 1980.

7.1. Database construction

- As usual all underestimated the work required to put together such a large and complex database. The text portion was more difficult that expected because of the fonts included; in addition, the librarians were very concerned with fonts and spacing being followed very exactly. The equations were complex because of the absence of standards for equations in some of the systems used (in particular ODA and SGML); as a result even in some systems of compound documents, the equations were displayed in image form. The figures were hard to extract accurately by automated means from the scanned images; it was often difficult to distinguish figures from equations, or to differentiate between one and two figures across a page.
- The use of a small database was invaluable in exercising the technology, learning to understand its limitations and gauging the extensions needed.

7.2. User access

- The use of a small document databases were invaluable for obtaining subjective feedback on what user facilities were required, and the relative advantages of the different types of user access.
- The three modes of access provided complementary forms of access: X-WAIS for content search on a single data base; PixLook with normal I-R search on the text portion of the database, and full access to an image form of the articles; SuperBook which allowed both conventional information retrieval and hypertext search. Of these all used text in the searching process—which could be done both over a LAN and remotely.
- For access to documents with mixed mode (e.g. SuperBook or XWAIS/ODA), the ISDN gives quite respectable performance. pre-fetching the complete paper improves this performance.
- For remote usage, the provision of small versions of diagrams, with the ability to request larger ones if desired, is very useful.
- Colour work stations are important in highlighting aspects of the searches; they are easier to use than monochrome ones.

7.3. Document formats

- Only SuperBook and XWAIS could realistically deliver the whole document remotely; the PixLook bitmap form was usually rather voluminous for extensive on-line perusal from outside a LAN (until SuperJanet is available!).
- It is inconvenient that we cannot store one form of database, and allow access by three different methods. Each access method requires a different form of database.
- The ODA form of document was the most convenient to incorporate into other documents. It was the only one in which the management aspects of the document are incorporated in the same database as the information itself. It is also the only one in which security features have been standardized.
- The SGML format is clearly the most appropriate for the publishers and can well incorporate full house styles; ODA is more suitable for blind reading of a

number of different databases. The lack of agreement on SGML DTDs is still a considerable nuisance—as we discovered in trying to use the UCL C-ODA software with the IOPP MSMSE journal.

- It was relatively easy to lay out the SGML into ODA once we ignored the problem of retaining the SGML structure for a subsequent conversion back into SGML. ODA is as good a choice for a presentation form as any other.
- Storing data in an ODIF form does not limit the user choice of tools. It can be used by any other editors which can read ODA documents. At the moment plenty such editors are available in the market.

7.4. User interest and facilities

- Users are much more interested in viewing documents from work stations in their vicinity than going any distance to a work station. For the UCL chemistry users this meant that at the least we needed to install Unix work stations locally. They would have preferred to use their own PCs or MACs from their offices.
- They are more comfortable in reading papers they really want on paper; we have not yet installed convenient printing facilities, but they are vital.
- The medium (10 000 document database) was the minimum size to allow chemists to really use the system—and even then their interest was limited. The principal bar to use was the limited number of years in the database. Unless there is a reasonable chance of the chemist finding the wanted references, there is little motivation to use the system.
- The ability to highlight through a search list is important. Viewing, editing and augmenting previous searches is important.
- There was considerable interest in the possibility of using the system to search automatically through references. This type of usage probably requires the full database.
- Chemists who are familiar with on-line databases are keen on registry numbers.
- In the image database, the use of the cruder 100 dpi size for browsing is convenient—but considered inappropriate for reading; proper text was considered better than image versions of it. Speed for images is important. Even 300 dpi was considered unusable for some pictures. With the XWAIS/ODA version, software limitations in the UCL software only permit 80 dpi for the diagrams and equations—but the picture has been converted, and then does not cause any complaints.

8. ACKNOWLEDGEMENT

We acknowledge the help given to the project by a number of people. David Golds did much of the work described here while he was leading the project; Michael Lesk (Bellcore) has been a major driving force both to the CORE and C-ODA projects; Lorrin Garson (ACS) has kindly allowed us to use the ACS data; Fred Friend (UCL library), Jill Bailey (UCL library), Janet Cropper (UCL library), chemistry users have been important in the trials; Peter Williams (Sterling Software and UCL) and Sammy Sameshima (UCL) have been instrumental in the OSISEC and DOCSEC work. We acknowledge the support of the British Library R and D Dept in supporting the C-ODA project.

REFERENCES

[1] Lesk, M. (1991) The CORE Electronic Chemistry Library *Proceedings of the ACM Special Interest Group on Information Retrieval Conference, Chicago 1991.*

[2] ISO (1986) *Information processing—Text and office systems—Standard Generalized Mark-up Language (SGML)*, IS 8879, International Organization for Standardization (ISO).

[3] ISO (1988) *Office Document Architecture (ODA) and Interchange Format*, IS 8613, International Organization for Standardization (ISO)

[4] Nelson, J. *et al.* (1991) The role of the PODA project in the adoption and development of ODA *Computer Networks and ISDN Systems*, **21**, 175–185.

[5] Golkar, S. *et al.* (1991), ODA activities at University College London, *Computer Networks and ISDN Systems*, **21**, 187–196.

[6] Kahle, B. (1989) *Wide Area Information Server Concepts.* Technical report, Thinking Machines Limited.

[7] International Organization for Standardization (1991) (DIS 10166) *Information Technology—Text and Office systems—Document filing and Retrieval (DFR).* International Organization for Standardization (ISO).

[8] Lesk, M. (1994) Electronic chemical journals, *Analytical Chemistry*, **66**, 14, 747A– 55A.

[9] Remde, J. R. *et al.* (1987) SuperBook: an automatic tool for information exploration -Hypertext? In *Proceedings of Hypertext 87,* Chapel Hill, NC, 175–188.

[10] Lesk, M. (1990) Images in document retrieval: extraction of figures from pages. *Proc. Anglo-French-US Conference on Image Storage in Libraries and Museums.* York.

[11] Hu, M. (1994) *An Intelligent Hypertext System*, PhD thesis, University College London.

[12] EWOS FOD26/CCITT PM2,(1990) Document Application Profile, Office document format profile for the interchange of enhanced function mixed content documents in processable and formatted form, EWOS.

[13] Montasser-Khosari, G. and Kirstein, P. T. (1994) *On-Line Access to Multimedia Documents*, BLRDD R&D Report 6139, London.

[14] SGMLS—derived from ARCSGML by James Clark (jjc@jclark.com) . (Available for anonymous ftp from ftp.ifi.uio.no [128.240.88.1] in the directory SIGhyper/SGMLUG/distrib)

[15] BBN (1990) *SLATE: Multimedia Document Communication System Reference*, Manual, Version 1.2, BBN, Boston, USA.

[16] Kille, S. E. (1993) *ISODE8*, Vol. 1: *Overview*, ISODE Consortium, London.

[17] Williams, P. *et al.* (1994) *The OSI Security Package: OSISEC User's Manual*, Release 2.3, UCL, London.

[18] Golkar, S. *et al.* (1990) *The Specification of Security Facilities for Securing Whole ODA Documents*, Task 2/2/6, UCL, London.

APPENDIX 1 AN EXAMPLE OF SGML STYLES

An example of style is given below; The full set of presentation attributes are available elsewhere [13].

```
:STYLE
No style name indicates set defaults
{  LI = 0;LeftIndent
FLO = 0;First Line OFFSET
RI = 0;RightIndent
FONTSIZE = 10pt;pt is optionalonly these units
¡ are valid.
FONT = Times-Roman;
FACE = Normal;
LINEWRAP = ON;
JUSTIFY = Full;
UNDERLINE = OFF;
GAP_ABOVE = 1 li;
GAP_BELOW = 0 li;
ORPHAN = 1;
WIDOW = 1;
COLUMN = 1;Number of columns
TABS = 1 in, 2 in, +0.5 in;Tabs at 1, 2, 2.5, 3, 3.5, .
PaperSize = A4;either a name or (x,y){mm|in}
RM = 1in;right margin
LM = 1in;Left Margin
TM = .75 in;Top Margin
BM = .85 in;Bottom Margin
}

STYLE title
{  JUSTIFY = Centre;
FACE = Bold;
FONTSIZE = 18pt;
}

STYLE subtitle : titleInherit values from title {
FONTSIZE = 14pt;Then assign new values.
}
```

FIGURE 1. An example of a style specification.

The following elements can only appear in the base style: