Integration of Microcomputer and Mainframe Information Systems†

WILLIAM G. TOWN

William Town Associates Ltd., 9 Peachcroft Centre, Abingdon, Oxon, England OX14 2NA

Received January 16, 1991

The age of the monolithic information system is past. Systems are becoming increasingly heterogeneous, both in terms of hardware and software. As open system standards are more widely accepted, corporate computer environments will increasingly evolve on the "mix and match" principle—the most appropriate tool, on the preferred hardware/software platform for any particular job, will be selected. Systems integrators, companies able to work with a wide variety of components from different and competing hardware and software vendors, will play an increasingly important role. The evolution of this new type of system is considered.

INTRODUCTION

Until recently the developers of chemical information systems have had a "chemical structure centered" perspective of the total information system requirements of their clients.[1-4] This has led developers to try to build monolithic chemical information systems by adding data and text-handling components around the chemical structure system core. In the last few years there has been a recognition that the wider needs of the organization cannot be satisfied by this approach. The application of free text search systems, database management systems (DBMS), and other information-handling tools more generally within the organization require a different perspective, which is no longer necessarily chemical structure centered. This shift of emphasis has in turn led to a demand for chemical structure information systems that can be integrated with the total information environment of the organization.

To understand this new environment, it is necessary to examine recent developments in computer hardware, operating environments, and application software, including the client/server model of distributed computing; consider the impact of standards in areas such as communications networks and protocols, document content architectures, database query languages, and chemical structure representations; explore the opportunities which graphical user interfaces offer and the role of the system integrator in a situation in which software tool kits are becoming the rule rather than the exception.

ENTERPRISE-WIDE NETWORKS

In retrospect, the 1980's will be seen as a brief hiccup in the evolution of corporate computing environments. During this period, the flight into personal computing, which occurred as personal computers gained in power and respectability, was the natural response by computer users to the frustration they felt as a result of central control of computer resources and the slow response of Management Information Systems (MIS) departments to their processing needs.

Whether the end user was controlled or not, standardized or not, computing in isolation worked well for a while as users and vendors developed software and refined hardware to turn early desktop curiosities into serious processing machines. When relatively few people within a company used a computer of any type, the issue of communication was a small one. Today, where, in many companies, almost all staff have access to a desktop computer or workstation, the benefits of linking these many, and often disparate devices, into a network are self-evident. From the users' perspective their networked computer can provide access to electronic mail, meeting calendars, centralized databases, and through gateways into public networks, public online databases. From the MIS perspective, the benefits of enterprise-wide communication include all of these and, in addition, central control of company-wide data files (including centralized backup), easier upgrades of user applications (by making the upgrade available on the network for downloading by each user), and reduced central support problems in general.

CLIENT/SERVER MODEL

As enterprise-wide networks develop and as the power of the desktop and mainframe computers continue to increase rapidly, a new model of computing is emerging: the client/server model. Client systems encompass PCs of all varieties—MS-DOS-based, OS/2-based, or Macintoshes—and workstations of choice—proprietary or UNIX. Servers cover even wider ground, from dedicated print, file, and network servers at one extreme to minicomputers or mainframe systems at the other. This new computing model can lead to large increases in productivity as more computing power is placed in the hands of the workers—those in the organization who are closest to the actual application of the information technology.

Three styles of client/server computing have been defined:[5]

Client/server computing splits the processing of an application between a front-end portion on a PC or workstation (which provides local data manipulation and maintains a user interface) and a back-end portion on a server (which handles database processing and number crunching).

Cooperative processing is similar but, in addition, spreads data for a given application across several systems, making use of all the computers on a network and making data available to any user connected to the network. Cooperative processing is less restrictive than the basic client/server style but is generally more expensive to implement.

Distributed computing allows an application to run on more than one system—a PC and a mainframe, for example.

Where the boundaries between the three computing styles should be drawn is debatable, and the three terms are increasingly being used interchangeably in the literature.

STANDARDS FOR NETWORKING AND DATABASE INTERROGATION

A prerequisite for the client/server model, in a heterogeneous corporate computer environment, is the ability to network together disparate hardware elements, and, therefore, this model is highly dependent on the development and wide adoption of standards. Fortunately, the International Standards Organization (ISO) has been laboring for many years on the Open System Interconnection (OSI) standard for communication linking heterogeneous computer systems.[5] The OSI standard has a layered architecture consisting of seven layers; each layer building on the functions of the layer below it. There are subgroups within the seven layers. The first three layers deal with the physical aspects of connectivity: wiring and cables, data encoding, addressing, and data management.

† This article is dedicted to Prof. Michael F. Lynch in celebration of 25 prolific years at the University of Sheffield.

0095-2338/91/1631-0176$02.50/0

Layers four, five, and six provide interoperability among different systems: session control, dialogue management, and format conversion. The highest layer, number seven, handles such distributed applications as file transfer, remote file access, and database management.

Standards have emerged over a period of time as each layer has been accepted: the Consultative Committee on International Telephony and Telegraphy (CCITT)[7] X.25 standard for communication across packet-switched data networks (layer two, data link layer) was one of the earliest (approved in 1976 and modified in 1980); the standard for electronic mail interchange, CCITT X.400 (established in 1984), was the first higher layer application to be approved; the file transfer access and management (FTAM) standard was approved in 1988 and other standards such as virtual terminal protocol (VTP), common application service elements (CASE), and job transfer and manipulation are in advanced stages of approval. The OSI standard has been widely accepted, and important efforts are underway to implement it (e.g., DECnet was scheduled to become fully OSI compliant in 1990).

Not all important standards in the area of distributed systems have originated from official standards organizations. The structured query language (SQL) for relational database management systems (RDBMS) is a de facto standard that has resulted from widespread acceptance of a standard promulgated originally by IBM. Nearly all RDBMS products provide an application program interface (API) or host language interface (HLI) that allows other software products to use their access and retrieval code directly. The RDBMS queries are expressed by using the SQL syntax. Thus, any front-end software which provides an SQL query generation capability may address a database running on an SQL server across a network.

STANDARDS FOR DOCUMENT EXCHANGE

An important aspect of a networked user environment is the facility it provides users to exchange messages and documents via the network and to use the centralized facilities in the corporate environment for document archiving and retrieval. In a heterogeneous environment, standards for document exchange are of fundamental importance. Document content standards were originally designed for the exchange of text between users of different word processing software packages. With the recent extensions to these standards, they are now sufficiently general to support documents containing ASCII text, bit maps, compressed images, and vector graphics. Extension of these document content standards to vector graphics and images is clearly essential if they are to have relevance to the problem of handling scientific documents containing chemical structure diagrams, graphs, and other diagrams. An example of these extended standards comes from IBM, which has recently published its Mixed Object:Document Content Architecture (MO:DCA),[8] an element of Systems Application Architecture (SAA), which has component specifications for graphics, presentation text, and images.

At present there is no clear "winner" in the field of document content standards. Manufacturers' own internal document architectures [e.g., Digital's Compound Document Architecture (CDA) and Digital Document Interchange Format (DDIF), IBM's Document Content Architecture (DCA), etc.] compete with international standards such as ISO 8613 "Information Processing—Text and Office Systems—Office Document Architecture (ODA) and Interchange Format (ODIF)",[9] which has been endorsed by the European Computer Manufacturers Association (ECMA). An international initiative, known as the Computer-aided Acquisition and Logistic Support (CALS)[10] initiative, originally based on U.S. military specifications, incorporates existing standards such as SQL, the Standard Generalized Markup Language (SGML),[11] Product Definition Exchange Specification (PDES),[12] and the Initial Graphics Exchange Specification (IGES).[13] SGML is also an ISO standard (ISO 8879) and has a related Standard Document Interchange Format (SDIF). Two international standards for image compression (CCITT Group 3 and Group 4) originated as facsimile transmission standards but are now part of most document content standards. These image compression standards, which achieve compression ratios of 10:1 and 20:1, respectively, are inadequate for large format drawings, and the CALS Tiling Task Group, among others, are addressing this problem and have proposed a Tiled Raster Interchange Format (TRIF).[14]

In the context of this paper it is not appropriate to consider each of these standards in detail. However, to give the "flavor" of a document content standard, a brief overview will now be given of the ISO 8613 standard which specifies conformance levels in each of the following areas:

document architecture: several levels of formatted form, processible form, and formatted-processible form

content architecture: several levels of formatted form, processible form, and formatted-processible form and for various content types

document profile: the description of the document (e.g., a title or an abstract), comprehensive but limited to document characteristics

interchange format: two levels—one comprehensive, the other limited and suitable only for either formatted or processible form but not for formatted-processible

Many vendors are now actively involved in completing ODA systems or completing ODA products. The major implementors include Apple, AT&T, BT, Bull, Digital, Fujitsu, Hitachi, IBM, ICL, Mitsubishi, Nixdorf, NTT, Oce, Oki, Olivetti, Philips, Siemens, Televerket, Toshiba, Unisys, and Xerox.

STANDARDS FOR REPRESENTING CHEMICAL STRUCTURES

In chemical information systems, chemical structures are usually represented by a connection table that may or may not contain two-dimensional coordinates or other optional information. The connection table is the computable form of the chemical structure (i.e., the form required to permit structure and substructure searching). Display of structure diagrams on the computer screen and output of the diagrams to a variety of printers are integral functions of a chemical information system, these outputs being derived directly or indirectly from the connection table. When, however, the chemical diagram is to be used as an image or vector representation (for example, as part of a scientific report), it is usual for the connection table to be converted into another form.

A number of de facto standards for representing vector images have emerged for use in personal computer application packages such as word processors, presentation graphics packages, and desktop publishing software. Structure editors may produce a number of graphical formats which enable the structure diagram to be imported into a variety of applications for further processing. Among these de facto standards are

Hewlett-Packard Graphics Language (*HPGL*): a vector representation originally developed as a plotter control language

Encapsulated PostScript (*EPS*): an extension of the page description language developed by Adobe Systems for printer control

Computer Graphics Metafile (*CGM*): an ISO standard vector representation and exchange format (ISO 8632)

which enable the vector representation of the chemical

structure diagram to be used in word processors such as WordPerfect, Microsoft Word, Lotus Manuscript, etc.; desktop publishing packages such as PageMaker and Ventura Publisher; and presentation graphics software such as Corel Draw, Lotus Freelance, and Harvard Presentation Graphics. As networked versions of these packages appear and versions operating on Digital VAX and IBM mainframes become available, they will also increasingly compete in the document exchange arena.

Within the Graphical User Interface (GUI) of Application Environments (AEs), such as Microsoft Windows or OS/2 Presentation Manager (PM), diagrams can be exchanged between applications either by cutting and pasting via the clipboard or directly through Dynamic Data Exchange (DDE). Using these techniques, it is possible to transfer both the graphical image of the chemical structure as public data and the underlying computable form (the connection table) as private data. Providing the application into which the diagram is cut and pasted preserves the private data intact, diagrams saved as part of a document could later be cut and pasted into front-end software, such as STN Express developed by Hampden Data Services for Chemical Abstracts Service (CAS), for use in a search of a public online database, such as the CAS Registry File (available from STN International and Questel) or Beilstein (STN, DIALOG, or Maxwell Online); into molecular modeling software, such as Alchemy from Tripos; or into a local structure database system, such as PsiBase from Hampden Data Services. The role of the GUI in providing a single user interface onto the whole world of information and corporate computing will be explained in a later section.

Standards are emerging for the representation of chemical structures as connection tables. Here, too, there is competition between vendors' "proprietary" formats [e.g., the Molecular Design Limited (MDL) Molfile] and emerging international standards, of which there are currently a number, including:

Standard Molecular Data (SMD) Format: developed originally by the CASP (Computer-Aided Synthesis Planning) consortium of European chemical and pharmaceutical companies and used increasingly by software vendors and database suppliers as an exchange format[15]

JCAMP-CS Format: developed by the Joint Committee on Atomic and Molecular Physics (JCAMP) as a complement of the JCAMP-DX format for spectroscopic data[16]

Standard Crystallographic File Structure: developed by a joint working part of the Data and Computing Divisions of the International Union of Crystallography[17]

It is likely that the present situation will be rationalized within the next few years. The SMD Subgroup of the Chemical Structure Association (CSA) is being supported by a large number of software vendors, database suppliers, and user companies (mostly chemical and pharmaceutical). Through its working parties, the SMD Subgroup is in discussion with the proponents of the competing standards in order to meet their requirements as part of this more general format. The American Society for Testing and Materials (ASTM) has decided to adopt SMD as the basis of a standard format for its own requirements. However, until that time comes, it will be necessary for applications software, such as PsiBase and STN Express, to be able to import and export a number of connection table formats (SMD, Alchemy Molfiles, DARC F1, etc.) as well as linear notations, such as SMILES (Simplified Molecular Input Line Entry System).[18]

GRAPHICAL USER INTERFACE STANDARDS

The subject of Graphical User Interfaces (GUIs) was alluded to briefly in the section above. The GUI has taken many years to evolve from its first inception at Xerox's Palo Alto Research Center (PARC), through its adoption for the Apple Lisa and later the Apple Macintosh, to its current widespread availability in a variety of forms on a wide variety of hardware platforms. The dramatic success of the Macintosh is in large part due to its user-friendly GUI supported by hardware designed for the job. Early attempts to emulate the Macintosh on IBM PCs and compatibles, in the form of the Digital Research GEM and Microsoft Windows GUIs, were only partially successful, being limited by the speed of the processor and (at the time) the low resolution of the graphics screen.

However, the recent release of Windows v.3 has at last brought the IBM "world" on a par with, or even in front of, the Macintosh "world" in terms of the features offered in the GUI. However, Windows v.3 does not merely boast an improved GUI—in all aspects this new application environment is a significant improvement over previous versions. The new Microsoft Windows version is bringing to the older PC-DOS/MS-DOS operating system many of the capabilities previously only offered in OS/2 through its Presentation Manager, a close relative of Windows. The inhibiting 640-Kbytes memory barrier inherent in PC-DOS has been broken in Windows v.3, which offers a virtual memory space of 16 Mbytes when operating in Standard or Enhanced Mode. In addition, true multitasking and the ability to run multiple DOS sessions simultaneously would in themselves be attractive to software developers and users alike even without the much-improved interface, which surpasses the Macintosh interface in some details.

Today, the Windows Icon Mouse Pointer (WIMP) GUI is so familiar to users that it is not necessary to describe it in any detail. For the foreseeable future, there will continue to be minor differences between the Macintosh, Windows, and Presentation Manager GUIs and also with other variants such as X-Windows, Open Look, NextStep, PM/X, and the Open Software Foundation's (OSF) Motif in the UNIX "world", DECwindows in the VAX VMS "world", and Hewlett-Packard's New Wave (based on Microsoft Windows). However, there should be enough commonality between these different GUIs to enable a user familiar with one to adapt easily to another.

There is a growing trend for software developers to provide versions of their application software packages for a number of GUIs, and although the additional expense of maintaining multiple software versions will inevitably be passed on to the user, the savings in user training and support costs are likely to be greater. Each GUI has its own idiosyncracies; for example, if an application is to be a true Macintosh application it must follow scrupulously the Apple Human Interface Guidelines. The new generation of chemical structure applications for these GUIs is beginning to emerge: Hampden Data Services have developed a Macintosh version of STN Express (Figure 1), which was released in 1989, and a Microsoft Windows version (Figure 2), released in 1991. Developers will endeavor to keep these different versions as close as possible to each other in functionality and appearance within the constraints imposed by the different GUI guidelines.

INTEGRATED DISTRIBUTED SYSTEM—A VIEW FROM THE DESKTOP

Windows-oriented GUIs are intended to present to the user the typical desktop on which any number of tasks may be in progress, some of which are in suspension while a more urgent task is attended to (Figure 3). In a GUI the empty desktop is, in fact, not a blank screen—there are usually one or two "icons" (e.g., a wastebin and a file) ready for the users attention. A more normal situation is one in which the user is faced with a number of open windows, each representing a task

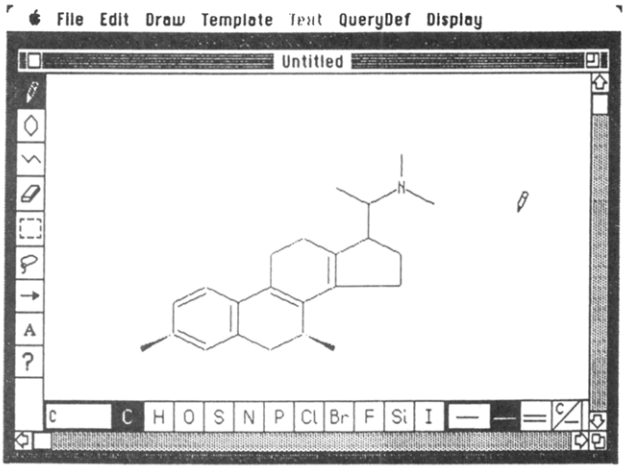

Figure 1. Chemical structure editor from STN Express for Macintosh.

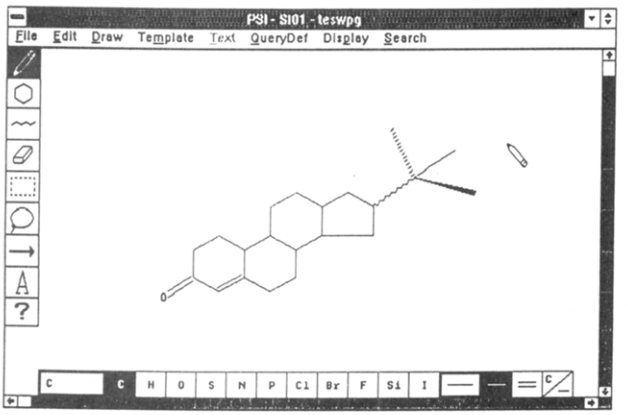

Figure 2. Chemical structure editor from STN Express for Windows.

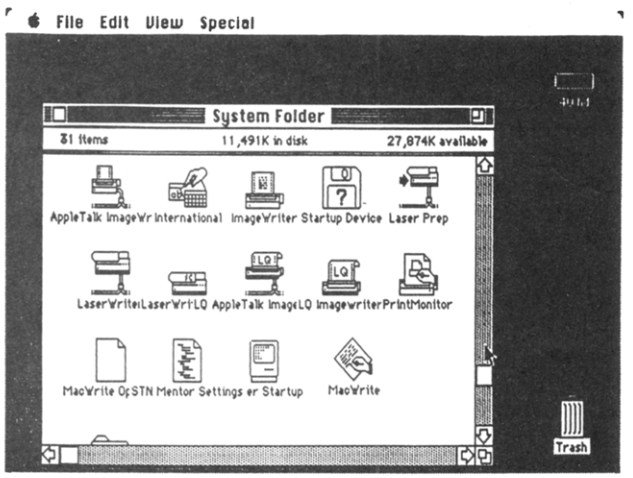

Figure 3. Macintosh GUI Desktop.

or application, overlapping each other like sheets of paper on the desktop and one or more icons representing tasks which have been "tidied away" to make more room on the desktop (Figure 4). Usually, only one of the tasks is active at a time from the point of view of user interaction, although for the other windows the task it represents may be continuing in the background. Any of the windows open on the desktop may be brought to the front (top of the desk) as the "active window" merely by pointing with the cursor to any part of it and clicking the mouse button. Windows may be moved around the desktop by pointing to the header bar and "dragging" with the mouse.

Information may be transferred from task to task by "cutting" (moving) or copying it first to a special window (usually invisible) known as the "clipboard", and then pasting

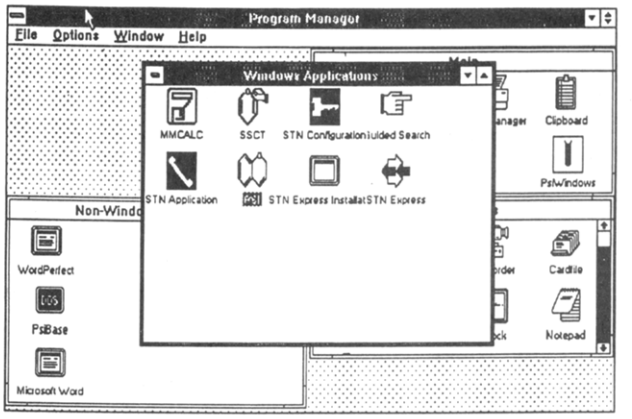

Figure 4. Microsoft Windows GUI Desktop.

the clipboard contents into the desired application or task. This is the more normal approach for, say, pasting a chemical structure diagram from a structure editor into a document being prepared in a word processor. It is also possible to exchange information dynamically between two applications. An example, frequently quoted, is the transfer of a table from a spreadsheet to a word processor using this technique. As the data contained in the spreadsheet is updated, the table in the document is automatically adjusted to reflect these changes. It is possible to have a number of these links active at a time.

Another example, more relevant to the topic of this paper, is a local structure database system. In such a system, the structure editor and the substructure search engine may be implemented as two separate tasks which communicate by Dynamic Data Exchange (DDE). The structure query is created in the structure editor and passed to the search engine when the menu button "Search" is selected. The results of the search may be passed back for review in the structure editor while the search is continuing in the search engine. Applications such as STN Express for Windows may consist of many tasks, each represented by a window or an icon on the screen. In STN Express, the terminal emulator, structure editor, text editor, and search assistant are all separate tasks which are normally initiated from the main task. If two applications are able to share information, it becomes possible to perform simultaneously a local database search and a search of one or more online databases such as the CAS Registry File or Beilstein with the same query.

To bring this discussion back into the context of the corporate-distributed computing environment, we need to consider which other tasks may be active on the desktop. Clearly, the mix of tasks is likely to change from time to time, but typically a user may also have a window open for electronic mail, windows for spreadsheets and word processors, perhaps reduced to icons, another window representing the front-end to an SQL RDBMS server running on a host on the corporate network, while a fifth may provide structural query access to the in-house structure and reaction databases. Add to these tasks the applications of local and public database access, with the ability to freely exchange information between applications by cutting and pasting or by dynamic data links, and the end user will experience a totally new work environment. The challenge is to provide integration of the type outlined in this section across networks of heterogeneous computers running a multiplicity of application environments.

TOOL KITS FOR BUILDING DISTRIBUTED CHEMICAL INFORMATION SYSTEMS

The foregoing sections have illustrated the complexity and the potential power of modern computing environments. It should be clear that no one supplier can hope to offer a

monolithic solution to the computing needs of an enterprise such as a modern pharmaceutical or chemical company, and as a result, strategic alliances are evolving. Increasingly, developers of chemical information systems are building tool kits[19] that enable system integrators to construct the "ideal" system for the particular corporate computing environment found in a given company.

In the "tool kit" concept, a "tool" fulfills a specific function, can be used independently, may easily be replaced by another tool as necessary, and is fully documented with respect to its function, implementation, and use. A simple example of a tool is a molecular formula search module, which may be implemented for use within a DBMS, is easily replaceable by another molecular formula search module, and whose molecular formula search capabilities and calls from any application are fully documented. In the DEC VAX/VMS environment, tools may exist in one of two forms: either a library of subroutines that can be linked to an application or as shareable images. They may be called from any programming language, from DBMS application builders, and from Digital's DECwindows application environment. Tools must be easy to set up, efficient with respect to response time and the use of computer resources, and easy to maintain leading to maximal use of the capabilities of the VAX/VMS operating system. Where possible, reentrant functions are used for optimization of oft-repeated tasks, and shareable images give independence between the tool and the application, ease of maintenance, and optimal performance. Some tools use interprocess communication. Analogous requirements apply to tool kits in other computing environments.

CONCLUSION

The enterprise-wide network, a prerequisite for fully integrated information systems, is increasingly becoming a feature of corporate computing environments. Network standards are facilitating the evolution of open systems built of heterogeneous components. New generations of information systems components based on de facto standard graphical user interfaces are creating new possibilities for integration. The requirements for document exchange and central archiving are being addressed by the developing standard document content architectures. The components are in place (or becoming available) that will enable the integration of corporate and public information systems to become a reality. A new breed of specialist (or, perhaps more correctly, generalist), the systems integrator, is evolving to meet the need.

REFERENCES AND NOTES

(1) Lynch, M. F.; Harrison, J. M.; Town, W. G.; Ash, J. E. *Computer Handling of Chemical Structure Information*; Macdonald/American Elsevier Computer Monographs; Macdonald: London, and American Elsevier: New York, 1971.

(2) *Chemical Information Systems*; Ash, J. E., Hyde, E., Eds.; Ellis Horwood Limited: Chichester, 1975.

(3) Ash, J. E.; Chubb, P. A.; Ward, S. E.; Welford, S. M.; Willett, P. *Communication, Storage and Retrieval of Chemical Information*; Ellis Horwood Limited: Chichester, 1985.

(4) *Chemical Structure Information Systems: Interfaces, Communication and Standards*; Warr, W. A., Ed.; ACS Symposium Series No. 400; American Chemical Society: Washington, DC, 1989.

(5) Francis, B. Client/Server: The Model for the '90s. *Datamation* **1990**, *36* (4), (Feb 15), 34–40.

(6) Bruce, J.; Kee, R.; Quinn, P. OSI (Open Systems Interconnection): The Commerical Benefit. *IES News* **1990**, *No. 30* (Oct), 19–21.

(7) CCITT Study Group recommendations are published on a 4-year cycle. The 1984 Recommendations (covering the 1981–1984 study period) known as the "Red Books" have recently been superseded by the 1988 Recommendations (covering the 1985–1988 study period) known as the "Blue Books" comprising some 18 500 pages. Further information from CCITT, Place des Nations, CH-1211 Geneva 20, Switzerland.

(8) IBM Reports: Data Stream and Object Architecture: Mixed Object Content Architecture Reference. Report SC31-6802; Data Stream and Object Architecture: Presentation Text Object Content Architecture Reference. Report SC31-6802; Data Stream and Object Architecture: Graphics Object Content Architecture Reference. Report SC31-6802; Data Stream and Object Architecture: Image Object Content Architecture Reference. Report SC31-6802.

(9) Carr, R. ISO 8613 Standard Permits Open Exchange of Documents. *IES News* **1989**, *No. 25* (Dec), 22–24.

(10) Holloway, H. CALS, SGML and All That Jazz! *Inf. Media Technol.* **1990**, *23* (4), 168–174.

(11) Federal Information Processing Standard 152.

(12) Proposed Federal Information Processing Standard.

(13) Proposed Federal Information Processing Standard.

(14) Anonymous. Progress Towards Tiling Standards. *Doc. Image Process.* **1989**, (May), 12.

(15) Barnard, J. M. Draft Specification for Revised Version of the Standard Molecular Data (SMD) Format. *J. Chem. Inf. Comput Sci.* **1990**, *30*, 81–96.

(16) Gasteiger, J.; Hendriks, B. M. P.; Hoever, P.; Jochum, C.; Somberg, H. JCAMP-CS A Standard Exchange Format for Chemical Structure Information in Computer Readable Form. *Appl. Spectrosc.* **1991**, *45* (1), 4–11.

(17) Brown, I. D. Standard Crystallographic File Structure. *Acta Crystallogr. A39*, **1983**, 216–224.

(18) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

(19) Huguet, P. The DARC Inhouse Packages as a Library of Standalone Functions for Building Applications in Handling Chemical Information. In *Proceedings of the 2nd International Conference on Chemical Structures: The International Language of Chemistry*; Warr, W. A., Ed.; Springer-Verlag: Heidelberg (in press).