J. Synchrotron Rad. (1994). **1**, 63–68

Computing for Synchrotron Radiation Experiments

K. S. Ackroyd, J. W. Campbell, C. E. Dean, M. Enderby, C. M. Gregory, M. A. Hayes, E. A. Hughes, S. H. Kinder, I. W. Kirkman, G. R. Mant, M. C. Miller, G. J. Milne, C. A. Ramsdale, P. C. Stephenson and E. Pantos*

DRAL, Daresbury Laboratory, Warrington WA4 4AD, UK

(*Received 25 March 1994; accepted 31 May 1994*)

Computing provision is of major concern to synchrotron radiation researchers. An overview is presented of the computing facilities for synchrotron radiation work at the Synchrotron Radiation Source, Daresbury Laboratory. Data acquisition, reduction and analysis are seen as an integrated activity essential for full utilization of beam time. We discuss the evolution of data-acquisition systems from stand-alone systems of limited computing power, to a unified project computing village using state-of-the-art equipment for combined-technique, high-data-rate experiments. A brief account of software packages for data reduction and analysis is also given.

Keywords: data acquisition; data analysis.

Introduction

Computing is an essential tool for synchrotron radiation research both for data acquisition and for data reduction and analysis. The requirement for remote control of experimental equipment and for handling of large data rates produced by, for instance, two-dimensional time-resolved experiments, places particular demands on data-acquisition and analysis software. At the same time, synchrotron radiation centres are multi-user, multi-experiment facilities, with a high throughput of projects and experimenters in a single year (for instance, over 500 experimental groups use beam time at the Daresbury Synchrotron Radiation Source, SRS, annually). This imposes certain constraints on the design and user interface of software packages as well as the organization of the corresponding infrastructure for hardware and software support; such constraints are not normally encountered in a single-group laboratory using software written by the experimenters themselves. We give here a very brief overview of software for data acquisition and data processing for synchrotron radiation at Daresbury Laboratory and the organizational issues involved.

Data acquisition

The rapid advance of computing technology has inevitably created a spread of technical solutions for data acquisition over the years. The first experiments at the SRS were controlled by 64 Kb-memory LSI-11 processors (Bordas *et al.*, 1986; Clark & Miller, 1990) of similar cost to present-day 32 Mb graphics workstations. We have learned the lesson that blanket solutions do not exist, or are even not desired. Slow-data-rate experiments or stand-alone 'laboratory bench' experiments are still perfectly adequately controlled by systems based on the successors to LSI-11 systems, MicroVAXs or PCs, using CAMAC or IEEE/488 interfaces to signal transducers and hardware controllers (Dent, Wells *et al.*, 1992; Oszlanyi & Miller, 1992; Kirkman & Buksh, 1992). This has the added advantage that, in some cases, user-specific software written at the home laboratory can be relatively easily ported or merged with the core codes at the SRS. New developments are spearheaded by the most demanding of applications, *i.e.* fast time-resolved, combined-technique and two-dimensional data-producing experiments, small-angle scattering (SAXS)/wide-angle scattering/differential scanning calorimetry, quick EXAFS/X-ray diffraction, small-angle diffraction/wide-angle diffraction, SAXS/Fourier-transform infrared spectroscopy (Dent, Wells *et al.*, 1992; Sankar *et al.*, 1993; Bras *et al.*, 1993; O'Kane *et al.*, 1994), some of which are controlled by VME-based systems networked to Unix workstations. In cases where the appropriate VME modules are not yet available, a hybrid approach of a VME processor interfaced to CAMAC has been used as an interim solution.

There is much scope for commonality within control software, *e.g.* the user interface, file I/O, graphics, motor driving, timing, pulse counting and memory use, with only the low-level code needing to be hardware dependent. Subroutine libraries, jointly developed and shared by project programmers responsible for the different stations, form the backbone of the data-acquisition programs, independent of operating system (DOS, VMS, OS-9) or platform (PC-AT bus, MicroVAX-Qbus or VMEbus). Development and maintenance can be carried out on appropriate platforms, usually Unix workstations, using a spectrum of software development tools (*CodeCenter*, *Dbx*, *FasTrak*, *FORCHECK*, *Lint*, *plusFORT*, *Purify*, *Quantify*, *RCS* and *StP*) each of

* To whom correspondence should be addressed.

which is suitable for different aspects of the software development cycle, *i.e.* specification and design, coding, testing, maintenance. Fig. 1 gives an example of the design of the top-level module for the EXAFS data-acquisition program as produced by the tool *StP* (*Software through Pictures*). *StP* uses detailed definitions of parameters and their logical connections to produce a consistent graphical presentation of the flow of data and control. Such diagrams help manage the layers of complexity and are most valuable at the very early stages of development, *before* any code is written, for validating the correctness of the specification and design. They form part of the program documentation (for the expert programmer and the specifier, not the user). Common developments of graphical user interfaces are coordinated across projects and make use of interface builders (*UIM/X*) for the design of *OSF/MOTIF*-based widgets. Such interfaces straddle the boundary between data acquisition and data analysis (Pantos, Dean, Stephenson, Milne & van Garderen, 1993).

An example of a package with wide application is the *PINCER* code (Miller & Ahern, 1991; Oszlanyi & Miller, 1992), written almost completely in ANSI C, which runs on both DOS and OS-9 based data-acquisition systems. Only the hardware I/O routines and the underlying graphics libraries are specific to each machine type. It currently operates on the X-ray diffraction stations 2.3, 8.4, 9.7, 16.3 and 16.4 controlling over a dozen hardware devices in five object classes. The code is developed and maintained under Unix with full version control and tested *via* a harness of hardware I/O simulation routines.

Hybrid VME-CAMAC and simple VME-only systems currently in use (energy-dispersive XAFS 7.4, 9.3, time-resolved X-ray diffraction 2.1, SAXS 8.2, surface diffraction 9.4, energy-dispersive powder diffraction 9.7) share low-level function libraries (Bogg *et al.*, 1992). More complex VME systems (16.1 time-resolved fibre diffraction, 16.2 X-ray diffraction from interfaces, 16.4 energy-dispersive powder diffraction, 16.5 ultra-dilute X-ray spectroscopy, 9.5 protein crystallography), share high-level generic objects as well as many of the low-level functions. Only the configuration and user interfaces vary across different stations to suit the different user communities, but even in these cases, house coding standards and style types are maintained to minimize the differences and maximize ease of use. Recent developments have led us towards client-server applications for motor control in which the server daemon runs on an embedded c.p.u. and communicates with the client using TCP/IP protocols over Ethernet. They are designed using object-oriented techniques and coded in C++. Object-oriented techniques enable faster response to advances in hardware and experimental techniques. From the software point of view, few cases are so unusual as to require totally experiment-specific code. The core code and utility functions remain common.

Fig. 2 shows a schematic of the data-acquisition system for the wiggler II station 16.1 for time-resolved fibre diffraction. VME modules, some with their own on-board processors, are controlled from the console of a SUN workstation on which *MOTIF* widget-based graphical user

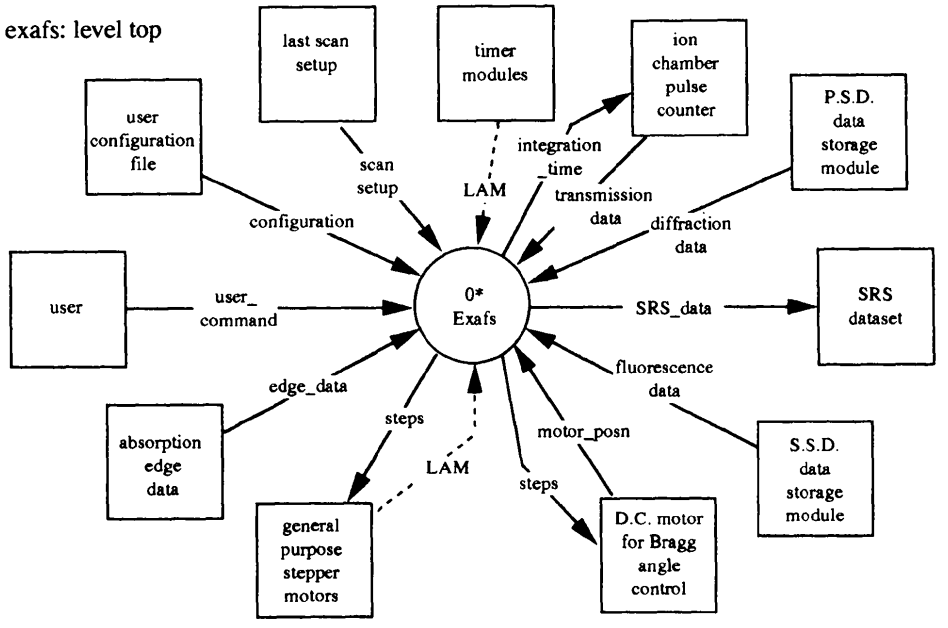

Figure 1
Top-level analysis diagram, or context diagram, for the data-acquisition program for the wiggler I station 9.3. The diagram was produced using the *StP* (*Software though Pictures*) *CASE* tool from IDE. This is the most abstract view of the system which is subsequently decomposed, using top-down analysis, into lower level diagrams. The tool performs automatic validation of the system analysis between and within levels. This process clarifies the analyst's understanding of the system and improves the integrity and efficiency of the analysis process by ensuring that the desired function is achievable from the given inputs.

interfaces provide for convenient user interaction. A local file server is used for immediate data storage. A dedicated fibre network connection gives fast access to other computing elements of the project's computing resources for further data reduction and analysis.

Data analysis

Data acquisition is, of course, not all that a synchrotron radiation experiment requires. On-line or off-line data reduction and analysis is increasingly an integral part of the activity during beam-taking shifts as some preliminary data processing is often necessary to ascertain data quality or to optimize experimental conditions. This is of particular importance to experiments producing large data sets, *e.g.* X-ray crystal or fibre diffraction using two-dimensional area detectors or image plates (Piltz, McMahon, Crain, Hatton & Nelmes, 1992; Martin-Fernandez, 1992). Analysis programs are typically run on workstations, some in the immediate locality of the experimental station, networked to the experiment control computer or the project file server.

The data analysis software collection available at the SRS includes programs available from other institutes (*e.g.* the powder diffraction program library *PDPL*, the protein crystallography data-reduction program *MOSFLM*, the simulated-annealing package *XPLOR*) or from the *CCP* (Collaborative Computational Project) collection, *i.e.* *CCP*3 (surface science), *CCP*4 (protein crystallography), *CCP*13 (fibre diffraction), *CCP*14 (powder diffraction). Other packages used predominantly for immediate data reduction and supported by in-house staff now constitute the major part of the SRS program library, much updated since the early days (Pantos, 1983). The one- or two-dimensional data analysis programs, *BSL*, *OTOKO*, *PLOTEK*, *MOTPLOT*, together with a number of utilities for data format and type conversion are available to all users, for use either on-site or at the home laboratory, and are implemented on a variety of platforms (SG, SUN, HP, DEC, IBM). Similarly, the XAFS analysis suite, *EXCURVE*92, *EXBACK*, *EXCALIB*, the fluorescence-lifetime analysis program *FLUOR* (Gregory, Hayes, Jones & Pantos, 1994), the SAXS simulation program *DALAI* (Pantos & Bordas, 1994) and the Daresbury Laboratory *LAUE* software suite (Helliwell *et al.*, 1989) are available to the academic community. In-house support for data analysis extends to the development of interfaces for public domain packages or those available through the UK CHEST organization (*e.g.* *EXPLORER*, *AVS*). Research projects in which computing staff are involved often lead developments in simulation and modelling software (Wade, Chretien & Pantos, 1990; Diaz, Pantos & Bordas, 1992; Andreu *et al.*, 1992; Diaz, Pantos, Bordas & Andreu, 1994; van Garderen, Pantos, Dokter, Beelen & van Santen, 1994), two-dimensional data analysis (Bordas *et al.*, 1993; Martin-Fernandez *et al.*, 1994), new Laue analysis methods (Campbell & Hao, 1993; Hao, Campbell, Harding & Helliwell, 1993), XAFS data reduction (Dent, Stephenson & Greaves, 1992) and parallel applications (Yeung, Pantos, Bordas & Diaz, 1992; Dean, Denny, Stephenson, Milne & Pantos, 1994).

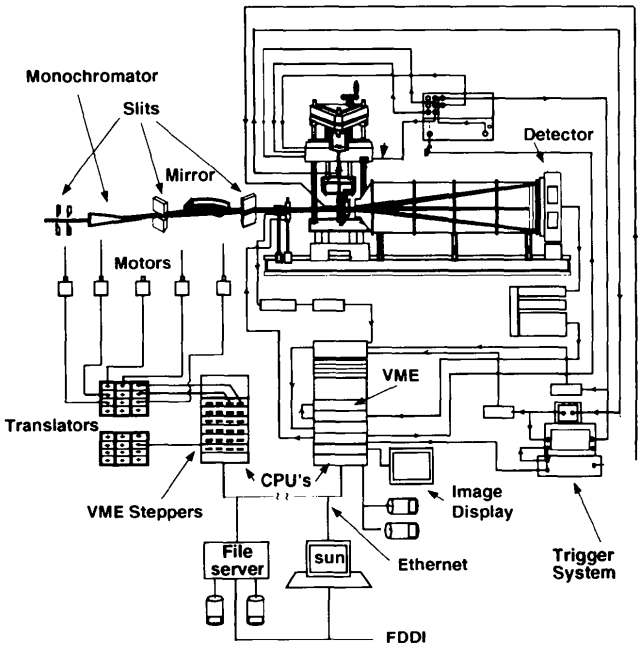

Figure 2
Schematic of the data-acquisition system for the wiggler II station 16.1 for time-resolved fibre diffraction. Control is based totally on VME modules some of which have their own on-board processor. The VME crate is interfaced *via* Ethernet to a SUN workstation on which user interaction is provided through *MOTIF* widget-based graphical user interfaces.

Organization of computing resources

The symbiosis of the multitude of computing systems of different type and manufacture targeted at different applications is not a totally straightforward matter. At the time of writing the SRS distributed computing system consisted of some 100 workstations, compute servers and file servers (SG, SUN, DEC, IBM, E&S and HP) located at the experimental stations, in the main computer room or in laboratories and offices. In addition, scores of IBM PC clones and Macintosh personal computers are scattered around the site, mainly in offices and laboratories. However, the predominantly Unix environment permits virtually seamless integration of resources, the main difficulties being of an administrative rather than technical nature.

The computing resources of each major science area are organized into 'villages'. This provides for administrative simplicity as resources can be augmented or upgraded as the need arises and finance permits. It also leads to the creation of a computing environment tailored to particular scientific needs. Homogeneity in manufacturer and operating system is obviously an advantage where it can be achieved. The key to developing resources to suit particular demands within a project is the flexibility of planning and control that the village concept offers.

Each project has at least one compute server for users working in that area. File services are provided centrally or from within the village, as required. A number of central servers handle general requirements (mail, printing, data archive and backup, terminal access) and access to/from the wide area network (WAN). The resources of each village are available at every desktop, including Macintosh or IBM PC clones connected to the laboratory local area network (LAN). File sharing between PCs and other computers (personal workstations or other shared resources) is accomplished *via* NFS or Xinet Kshare. This also allows access to other peripherals such as tape drives and printers. Access is by terminal emulator or X-windows. All systems are connected to the LAN and through it to central services and the outside world.

The wide geographical distribution of project resources does not permit their grouping into actual village LANs bridged to the site network but only into 'logical villages'. High throughput fibre network connections (FDDI) are introduced as the need dictates, for example, between file and compute servers and stations which produce high data rates. Fig. 3 gives an example of a computing village, the one for protein crystallography, showing only the key elements of the total resources, for both data acquisition and immediate data reduction, *e.g.* image-plate data readout and correction, as well as for off-line data analysis.

An application server hosts in-house developed or public domain and commercial packages of cross-project interest. Molecular modelling and two- or three-dimensional image processing packages (*INSIGHT*II, *FRODO/O*, *SIRIUS*, *SPIDER*, *VoxelView*) utilize high-performance graphics workstations (E&S PS300, ESV10, SG Indigo^2 Extreme or Indigo Elan) dedicated to specific projects. Recent developments with software harnesses for parallel program execution on clusters of networked computers permit the *ad-hoc* organization of village computing resources into 'parallel virtual machines' whereby CPU-intensive applications (Dean *et al.*, 1994) can utilize spare compute capacity on a variety of machine types without undue disruption to their normal use.

The future

We expect the overlap between data-acquisition software, electronics modules with on-board processing power and on-line data analysis packages to become much greater with the boundaries becoming less well defined. Technical advances in synchrotron radiation sources and instrumentation, particularly fast detectors, are producing major challenges to the ingenuity of software engineers and infrastructure services. There is no technical impediment, at least for some cases, as to why multiprocessor machines of the distributed or shared memory type should not reside in the experimental control rack with multi-dimensional data collected, reduced and analysed on-the-fly, as the photons strike the detector. Neural-net and fuzzy-logic based expert systems with voice I/O as well as more sophisticated three-dimensional simulation software are very likely to become part of the feedback loop between raw data collection and data interpretation. The partnership between software engineer and scientist is going to be crucial in tackling these challenges.

Conclusions

Computing developments continue apace, surprising even the professionals. The days of a single laboratory mainframe accessed by 'dumb' terminals are gone and the age of distributed, fibre-networked, harnessed-in-parallel computing resources are upon us. High-performance computing is in a phase of rapid evolution. Already, mainframe-on-the-desk power is widely available, blurring the distinction between the so-called desktop personal computers and powerful graphics workstations. Network management tools are an increasingly important element of infrastructure as distributed computing resources increase in number, power and sophistication.

Despite the plethora of high-quality, commercially available or public domain software, the need for development of specialized codes for synchrotron radiation work will remain strong. Plentiful computing power is certainly enabling more ambitious approaches to problems, old and new, but that alone is not sufficient. Provision of raw computing power is becoming a matter of small-scale finance relative to the investment required for the experimental facilities themselves. All the same, hard technical problems still remain to be tackled. Specialized software development will remain labour-intensive requiring expert interdisciplinary skills. Software design and production tools, object-oriented programming techniques and graphical user interfaces require skills that full-time synchrotron radiation scientists usually do not possess. Software packages are becoming more complex as well as more powerful and versatile, putting higher demands on software specialists.

The authors wish to acknowledge the support of their colleagues in the Computing and Electronics Division and the SRS users and scientists for stimulating their work. We also thank Dr Marissa Martin-Fernandez for drawing Fig. 2.

References

Andreu, J. M., Bordas, J., Diaz, J. F., Garcia de Ancos, J., Gil, R., Medrano, F. J., Nogalez, E., Pantos, E. & Towns-Andrews, E. (1992). *J. Mol. Biol.* **226**, 169–184.

Bogg, D., Dent, A. J., Derbyshire, G. E., Farrow, R. C., Greaves, G. N., Helsby, W. I., Morrell, C., Ramsdale, C. A. & Wells, M. P. (1992). *Rev. Sci. Instrum.* **63**, 790–791.

Bordas, J., Boulin, C., Gabriel, A., Mant, G. R., Mochiki, K., Sumner, I. & Worgan, J. S. (1986). *Nucl. Instrum. Methods*, A**246**, 545–548.

Bordas, J., Diakun, G. P., Diaz, F. G., Harries, J. E., Lewis, R. A., Lowy, J., Mant, G. R., Martin-Fernandez, M. L. & Towns-Andrews, E. (1993). *J. Muscle Res. Cell Motil.* **14**, 311–324.

Bras, W., Derbyshire, G., Ryan, A. J., Mant, G. R., Felton, A., Lewis, R., Hall, C. & Greaves, G. N. (1993). *Nucl. Instrum. Methods*, A**326**, 587–591.

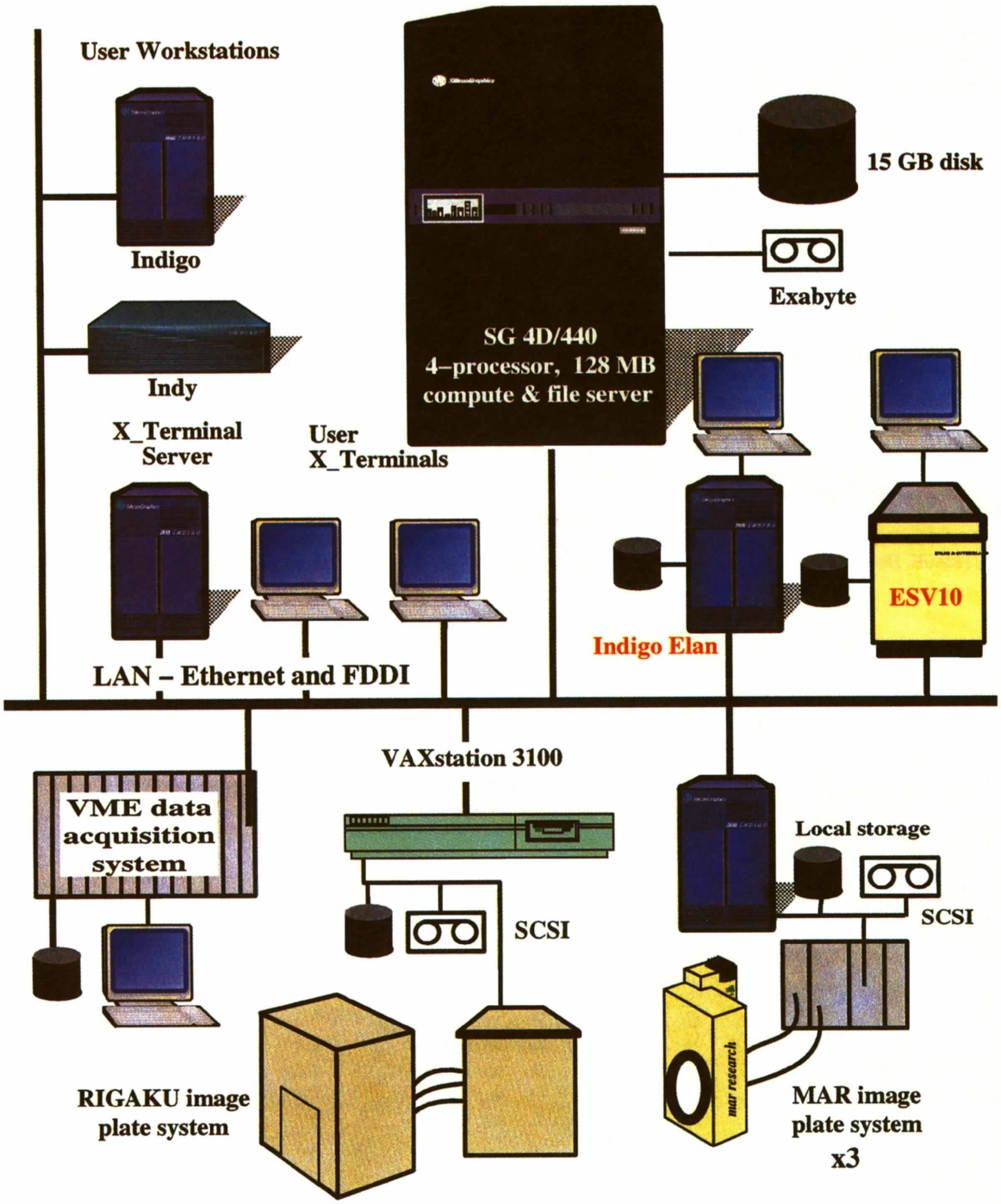

Figure 3
Schematic of the computer village for protein crystallography. The top–bottom division reflects the analysis–acquisition allocation of resources linked by the local area network which provides access to all other site services and the rest of the world through the wide area network. The village serves the data-acquisition and data-reduction needs of stations 7.2, 9.5 and 9.6, software maintenance and development (including *CCP*4), and in-house research and development.

Campbell, J. W. & Hao, Q. (1993). *Acta Cryst.* A**49**, 889–893.
Clark, S. M. & Miller, M. C. (1990). *Rev. Sci. Instrum.* **61**, 8.
Dean, C. E., Denny, R. C., Stephenson, P. C., Milne, G. J. & Pantos, E. (1994). *International Conference on Frontiers of Synchrotron Radiation*, Aix en Provence, France, 5–12 April 1994.
Dent, A. J., Stephenson, P. C. & Greaves, G. N. (1992). *Rev. Sci. Instrum.* **63**, 856–858.
Dent, A. J., Wells, M. P., Farrow, R. C., Ramsdale, C. A., Derbyshire, G. E. & Greaves, G. N. (1992). *Rev. Sci. Instrum.* **63**, 903.
Diaz, F. G., Pantos, E. & Bordas, J. (1992). *Rev. Sci. Instrum.* **63**, 859–862.

Diaz, J. F., Pantos, E., Bordas, J. & Andreu, J. M. (1994). *J. Mol. Biol.* **238**, 214–225.

Garderen, H. F van, Pantos, E., Dokter, W. H., Beelen, T. P. M. & van Santen, R. A. (1994). *Model. Simul. Mater. Sci. Eng.* **2**, 295–312.

Gregory, C. M., Hayes, M. A., Jones, G. R. & Pantos, E. (1994). *FLUOR. A Program to Analyse Fluorescence Data.* Technical Memorandum DL/SCI/TM98E. Daresbury Laboratory, Warrington WA4 4AD, UK.

Hao, Q., Campbell, J. W., Harding, M. M. & Helliwell, J. R. (1993). *Acta Cryst.* A**49**, 528–531.

Helliwell, J. R., Habash, J., Cruickshank, D. W. J., Harding, M. M., Greenhough, T. J., Campbell, J. W., Clifton, I. J., Elder, M., Machin, P. A., Papiz, M. Z. & Zurek, S. (1989). *J. Appl. Cryst.* **22**, 483–497.

Kirkman, I. W. & Buksh, P. A. (1992). *Rev. Sci. Instrum.* **63**, 869–872.

Martin-Fernandez, M. L. (1992). PhD thesis, Univ. of Keele, UK, and Daresbury Laboratory, UK.

Martin-Fernandez, M. L., Bordas, J., Diakun, G., Harries, J., Lowy, J., Mant, G. R., Svensson, A. & Towns-Andrews, E. (1994). *J. Muscle Res. Cell Motil.* In the press.

Miller, M. C. & Ahern, S. (1991). *Specification for SRS Wiggler-2 Station 16.3 Data Acquisition System.* Technical Memorandum DL/SCI/TM80E. Daresbury Laboratory, Warrington WA4 4AD, UK.

O'Kane, W. J., Ryan, A. J., Young, R. J., Bras, W., Derbyshire, G. & Mant, G. R. (1994). *Polymer*, **35**. In the press.

Oszlanyi, G. & Miller, M. C. (1992). *A Novel Data Acquisition System Based on a General Purpose Control Language.* Technical Memorandum DL/SCI/TM86E. Daresbury Laboratory, Warrington WA4 4AD, UK.

Pantos, E. (1983). *Nucl. Instrum. Methods*, **208**, 449–459.

Pantos, E. & Bordas, J. (1994). *Pure Appl. Chem.* **66**, 77–82.

Pantos, E., Dean, C. E., Stephenson, P. C., Milne, G. J. & van Garderen, H. F. (1993). *Parallel Implementation of the Debye Formula for the Calculation of the Small-Angle X-ray Scattering of Macromolecular Structures.* Preprint DL/CSE/P27E. Daresbury Laboratory, Warrington WA4 4AD, UK.

Piltz, R. O., McMahon, J., Crain, J., Hatton, P. D. & Nelmes, R. J. (1992). *Rev. Sci. Instrum.* **63**, 700–703.

Sankar, G., Wright, P. A., Natarayan, S., Thomas, J. M., Greaves, G. N., Dent, A. J., Dobson, B. R., Ramsdale, C. A. & Jones, R. H. (1993). *J. Phys. Chem.* **97**, 9550–9554.

Wade, R. H., Chretien, D. & Pantos, E. (1990). *Proceedings of NATO Workshop on Electron Crystallography*, Erice, edited by J. R. Fryer & D. L. Dorset, pp. 317–325. Dordrecht: Kluwer Academic Publishers.

Yeung, H. K. F., Pantos, E., Bordas, J. & Diaz, F. G. (1992). *Parallel Computing and Transputer Applications*, edited by M. Valero, E. Onate, M. Jane & B. Suarez, pp. 1070–1079. Barcelona: IOS Press/CIMNE.