BCIS

Biodiversity Conservation Information System

Framework for Information Sharing
Series Editor John R. Busby

Volume 8: TOOLS & TECHNOLOGY

Citation: Biodiversity Conservation Information System. 2000. *Framework for Information Sharing: Tools & Technology* . Busby, J.R. (Series Editor).

Available from Program Manager, BCIS [contact details on http://www.biodiversity.org]

Every reasonable care has been taken to check that all web addresses are correct at the time of writing (December 1999). Regrettably, sites and pages change over time so some addresses may no longer be correct.

Vol. 8: Tools & Technology ◆ Contents

Acknowledgements

These handbooks have been loosely modelled on the *WCMC Handbooks on Biodiversity Information Management*, edited by Jake Reynolds of the World Conservation Monitoring Centre and published by the Commonwealth Secretariat, London. These, in turn, benefited from experiences gained through the Biodiversity Data Management (BDM) Project, administered by the United Nations Environment Programme (UNEP) and funded by the Global Environment Facility (GEF), and related initiatives supported through the European Union (EU) and European Environment Agency (EEA).

Fundamental to the development of this series have been the contributions from numerous colleagues working in the field of biodiversity information management. Among these, particular mention goes to Dr Jake Reynolds and other staff of WCMC; Professor Ian Crain and Gwynneth Martin of the Orbis Institute, Ottawa; Kevin Grose, former Head of the Information Management Group at IUCN–The World Conservation Union; and staff of the Environmental Resources Information Network (ERIN) Unit, Environment Australia, and of the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia.

This Handbook has benefited from World Conservation Monitoring Centre. 1998. WCMC Handbooks on Biodiversity Information Management. Volume 7: Data Management Fundamentals. Reynolds, J.H. (Series Editor). Commonwealth Secretariat, London. ix + 33pp. Heavy use has also been made of dictionaries of information technology, in particular Shnier, M. (1996). *Dictionary of PC Hardware and Data Communications Terms* and Howe, D (ed.) (1998) *The Free On-line Dictionary of Computing.*

The BCIS consortium Members extend their thanks and gratitude to the Norwegian Agency for Development Cooperation (NORAD) for providing financial support to develop the BCIS concept and these handbooks.

BCIS Framework for Information Sharing

The purpose of the *BCIS Framework for Information Sharing* is to support BCIS Members and others making decisions on the conservation and sustainable use of living resources. The handbooks form part of a comprehensive set of supporting materials designed to build information-management capacity and improve decision-making.

The intended audience includes senior managers in Member organisations, their equivalents in other organisations, information and environmental-science professionals, and others who have a stake in the use or management of living resources. Although written to address the specific need for improved management of biodiversity-related information within the BCIS network, the underlying principles apply to environmental information networks in general, and to decision-making at all levels. The issues and concepts presented may also be applied in sectors other than biodiversity conservation: forestry, agriculture, wildlife management and beyond.

The handbooks deal with a range of issues and processes relevant to the use of information in decision-making, including the strengthening of organisations and organisational linkages, data custodianship and management, metadata and the development of infrastructure to support data and information exchange. Experience suggests that some of the greatest challenges in information management today are concerned with organisational issues, rather than technical or scientific concerns. Consequently, topics are addressed at management and organisational levels, rather than from a technical or scientific standpoint. Nevertheless, in adopting this framework approach, BCIS has tried to adhere to recognised conventions and formalisms used in information management.

Overall, the handbook series comprises:

Executive Overview

- Volume 1: Principles
- Volume 2: Procedures Manual
- Volume 3: Custodianship
- Volume 4: Data Access
- Volume 5: Metadata
- Volume 6: Standards & Quality Assurance
- Volume 7: Core Datasets
- **Volume 8: Tools & Technologies**

Collectively, the handbook series promotes a shift from tactically based information systems, aimed at supporting individual projects, to strategic systems that promote the development of information infrastructure through the building of capacity within BCIS and other networks. This approach not only encourages data to be managed more effectively within organisations, but also encourages data to be shared amongst organisations for the development of the integrated products and services needed to address complex and far-reaching environmental issues.

The handbook series can be used in a number of ways. Individual handbooks can be used to guide managers and professional staff on specific aspects of information management or they can be used collectively as a reference source for strategic planning and project development.

The Handbook reviews issues associated with the selection and maintenance of the technologies necessary to support data management and the generation of information products. It covers technology selection, software management, data and information management, analysis and modelling tools, and information dissemination.

Manager's Guide

Context

Information and communication technologies have now become indispensable to the management, analysis and dissemination of data and information. They can dramatically improve the cost-effectiveness of many projects, which themselves can be expensive. On the other hand, if managed inappropriately, they also have the potential to raise costs, disrupt work patterns and generally inhibit project execution.

The proper role for information technology is to support, *but not drive,* corporate information management objectives. Difficulties arise where technology is permitted to drive strategy, usually because managers lack the knowledge or confidence to give clear direction and exercise appropriate control. Inappropriate delegation of responsibility to technical-oriented staff can result in expenditure on hardware systems that are more elaborate and expensive than necessary, software systems that are inappropriate to support key corporate objectives and applications systems that do not meet corporate or other user needs.

Inevitably, the introduction of information technology involves new costs, not only the cost of purchase but also that of maintenance, training and support. Technologies age very rapidly, so all these costs are ongoing. Another challenge for management is to strike the optimal balance between investment in internal infrastructure and outsourcing tasks, where this is cost-effective. Certain core competencies will need to be developed and retained 'in house'. Information systems, of course, need to evolve as the information needs of an organisation change.

Actions

Managers need to ensure that information technology supports core business objectives and does not become an end in itself. Investments made in information technology need to be matched by investments in appropriately trained staff. These staff need to be managed to ensure that technology acquisition and upgrade programs empower their remaining staff to meet corporate objectives, and are cost effective and sustainable.

Different technologies advance or, conversely, age at different rates, so a rolling upgrade program should be put in place. The key here is to judge the pace of change and try to move neither too fast (and risk being stranded by transient fads) nor too slow (and risk being stranded with legacy equipment with difficult upgrade paths). Increasingly, perhaps, hardware should be regarded as consumable rather than capital items, with perhaps more attention being paid to ensuring support for business objectives, upgrading policies and procedures, and staff hiring and training.

Even where substantial proportions of technology-based applications are sourced externally, it is important that the agency hold sufficient technical competence to actively manage the process. Responsibility should never be abdicated to suppliers or contractors and there need to be clear specifications on what is to be delivered. There should also be sufficient technical capacity to ensure compliance with the specifications and ensure that the technology integrates smoothly and effectively with corporate operations.

1. Introduction

Information and communication technologies are extremely powerful and can be highly cost-effective. In fact, an increasing majority of data management and information product generation activities would be literally impossible without them. Systems can store huge volumes of data and perform complex analyses very rapidly. They can also validate data as they are entered, store them in a variety of forms, and produce many different kinds of reports, charts, maps and other products from the same data. They can thus dramatically improve the cost-effectiveness and enhance the range and comprehensiveness of data-based projects.

Each organisation needs to develop its own information strategy, comprising clear aims and objectives for the implementation and performance auditing of information technologies. The consequences of not doing this, of allowing the agency to drift or abdicating responsibility to others, can be very severe. The organisation's performance will almost certainly suffer. Its survival might even be at risk.

A key issue is achieving the right balance between building internal technological infrastructure and outsourcing tasks to partners or to others. Many routine tasks, such as corporate accounts and payroll, are common to all enterprises and have well-developed technology solutions. It can be cost-effective to purchase such services from specialist external sources. Other high-cost or very specialised applications, such as designing pages for a world wide web site or satellite image processing, may not warrant the hiring of the specialist skills required and may be better sought externally.

Outright purchase of services from an external supplier can, of course, be costly. Increasingly, agencies are looking to build collaborative arrangements with others to share or partition the cost of infrastructure development and to trade or barter goods and services. Providing all the necessary arrangements have been agreed, usually through formal contracts or at least written agreements, and every player knows the 'rules', this can be highly cost-effective.

Regardless of the extent of sharing or outsourcing, agencies will still need to plan and develop their own data management infrastructure. There will be certain core competencies that an agency will wish to develop or enhance. The following sections discuss the issues and options involved.

2. Selecting Technology[1]

2.1 Issues to Consider

There are a number of issues to consider in selecting the technologies that are most appropriate to meet corporate and user needs. Some of the most important are:

Scalability

As the number of users, records or attributes grow, an application that once performed well on a low-cost computing architecture can deteriorate in performance quite dramatically, to the point where it is barely functional. Typically, stand-alone or small network computer architectures are most likely to suffer from this problem, which explains the rise of more sophisticated architectures, such as client-server.

Connectivity

To enable rapid exchange of data between individuals and organisations, electronic connectivity is essential. This could take the form of a group of locally-networked computers sharing a common storage area (e.g. a file server), or more sophisticated dial-up communication lines to external services, such as the Internet and private networks. The capacity to connect computers together into more powerful resources is increasingly recognised as the key to rapid access and use of data.

Compatibility

The issue of compatibility is diminishing as manufacturers evolve a range of standard specifications for their IT products. However, the specifications—which are often proprietary in nature—are still too varied and numerous to discount the problem entirely. As far as computing platforms are concerned (i.e. computer hardware plus operating system), major decisions include whether to adopt IBM-PC-compatible computers running derivatives of the Microsoft Windows® operating system, or larger workstations running Unix. Since the technologies are changing so rapidly, there is really no 'best' solution, other than to adopt a platform which has proved to be reliable and useful in circumstances similar to those antici-

[1] Volume 7: Data Management Fundamentals. Reynolds, J.H. (Series Editor). Commonwealth Secretariat, London. ix + 33pp.

pated, working on the principle that, in such cases, compatibility issues are unlikely to cause serious disruption.

Sustainability

For information technology to deliver long-term improvements in effectiveness, sufficient funds and expertise must be available for users to exploit its potential fully and not be disadvantaged by its costs in terms of training, technical support and maintenance. Technology which has proven effective under the prevailing conditions is usually the best choice. The capacity of local suppliers to support the technologies and the potential for obtaining helpful advice from other nearby sites with similar systems also needs to be considered.

2.2 Evaluating Options

Following an analysis of needs and the issues outlined above, the advantages and disadvantages of different technological solutions should be tested under realistic local conditions before procurement takes place. Although useful, manufacturers descriptions, magazine reviews and specialist information services (e.g. Internet newsgroups and bulletin boards) should not be relied upon for strategic procurement decisions.

It should be noted that some characteristics of information technology are subjective, such as the ease of use of a software package or the quality of a scanned image. Thus, selecting technology purely from a list of features is unlikely to be satisfactory. A real-life test is the best way of determining whether technology will be suitable under the expected working conditions.

A wide range of options exist for managing data. These include single (stand-alone) computers running local copies of data-management software; locally-networked computers with shared software running on a file server (i.e. a Local Area Network or LAN); client-server architectures which integrate the best characteristics of personal computers (friendly software and quick response) with the best traits of file servers (high storage capacity, fast data processing, good security); and fully-distributed databases consisting of a series of remote computers linked via permanent or dial-up communication lines (i.e. a Wide Area Network or WAN). The decision as to which option to select should be taken after a thorough examination of the factors outlined above. Clearly, the nature and extent of the data to be stored will influence this decision greatly, as will the degree to which the data need to be accessed electronically by internal and external users.

2.3 Hardware Replacement Policies

It is important to develop a consistent rolling program of hardware upgrades. The key here is to judge the pace of change and try to move neither too fast (and risk being stranded by transient fads) nor too slow (and risk being stranded with legacy equipment with difficult upgrade paths).

Continually cascading the technology is generally good practice. The most mission-critical and demanding applications should have top-of-the-line hardware allocated to them. Hardware that has been overtaken, but is still effective, can then be allocated to less demanding tasks. Hardware at the end of the cascade should be disposed of. Increasingly, perhaps, hardware should be regarded as consumable rather than capital items.

Different hardware components age at different rates. Monitors, for example, depreciate far more slowly than processors, memory or hard disks. Investment in good quality monitors is therefore worthwhile.

3. Software Management[2]

Once the basic system has been selected and installed, along with its core operating system, then additional software will need to be obtained to manage, analyse and disseminate data and information. There are some important issues in software management, irrespective of the nature of software employed.

3.1 Software Licensing

The types of licensing available fall into the following categories:

3.1.1 Commercial Software

You must buy it or, perhaps more accurately, a licence to use it. Ordinarily this software is the most robust and reliable and generally comes with user support and regular upgrades. These features are not inexpensive. Developers need to cover all the development, testing, marketing, user support and other costs, not to mention profit, in part to cover past failures and to invest in future products. The higher the market penetration, the lower the overhead costs that need to be recovered from each unit sold and thus, in general, the cheaper the software. For general-purpose applications like word processing, spreadsheets and simple databases, commercial software is almost invariably the best option. For more specialised applications, however, such software may not or only partially fulfil the requirements. In these cases other options may need to be considered.

3.1.2 Freeware

"I wrote it, but you can have it". The software may be good or bad, upgraded regularly or a one-off, but it's free. The main restriction is basically to keep the freeware label on the package and not to charge somebody else for the software if you pass it on. Occasionally the software itself is free, but a charge is made to cover packaging, marketing and distribution costs. Usually there is no user support, although it may be possible to join an informal user group to exchange experiences and seek advice. One source of such software is the GNU Project of the Free Software Foundation <http://www.gnu.org/>.

---

[2] Derived from the Environmental Resources Information Network (ERIN) <http://www.environment.gov.au/epcg/erin/guidelines/technical/web_setup/sw_maint/sw_maint.html>

3.1.3 Shareware

Try it out, and if you want to keep using it, pay a registration fee. It can be a lot cheaper than commercial software. Sometimes the fee is only demanded for commercial use of the software, personal use is free. You need to read the license documentation to find out. You can also expect the quality of shareware to be a bit better than general freeware.

Freeware and Shareware fall under the general heading of Public Domain software.

3.2 Issues in Software Selection

When selecting software you need to decide on whether to use commercial or public domain software. Some points to consider are:

3.2.1 Software Policy

The policy of your organisation, with regards to public domain software, needs to be considered. Your organisation may not allow the use of public domain software, or there may not even be a policy. If not, one must be developed. Public domain software is handy for day-to-day tasks, but there could be risks with sensitive or mission-critical information unless you are confident you know what it is doing.

3.2.2 Budget Considerations

How much can you afford to spend? Public domain software may be free or require a registration fee, but even this fee is generally cheaper than investing in a commercial product. On the other hand, what is the cost if your bargain basement software munches up your system or your staff spend large amounts of time troubleshooting problems?

3.2.3 Reliability

There is good and bad in both the commercial and public domain. Look for a well renowned product, don't drag something from obscurity. You would expect the commercial versions to maintain user support, although most mainstream public domain software has supporting internet discussion groups anyway. Public domain software would take up more of your time if it came to narrowing down the cause of a fault.

3.2.4 Warranty

If software does something unexpected, particularly if it causes damage, "Where can you point the finger?" Commercial products should have user support to cover problems, but if it's from the public domain, its YOU who put it on the system. Most, if not all, software has a disclaimer, in effect saying that the authors will not be liable for any damage caused by using their software on your system. It's just possible that commercial vendors may be motivated to help you recover, to avoid bad publicity and so you'll continue to purchase their products.

3.2.5 Upgrades

Will the software be a longer-term solution? Can you expect updates that will keep the software on par with changes to your system? If the author of a piece of public domain software falls off a cliff, he may not be the only one at a dead end. There is a greater sense of security when you know some company or group is looking after the software, rather than an individual. This applies particularly to software developed by specialist groups for their own applications. Once they have achieved their objectives, they may lose all interest in the software and you may be left stranded.

Commercial software usually provides regular version upgrades, which are nearly always worth using. Public domain software, on the other hand, may fire off sets of patches, and give you versions to 100 decimal places. Its always tempting to grab every latest release and recompile, but it is just too time consuming. Apart from security patches and patches for problems you actually need fixed, you don't necessarily have to upgrade immediately.

3.2.6 Intellectual Property Rights

Whenever you download and install a public domain package, you need to check the license and copyright information that comes with it—usually in files with names like LICENSE, COPYING, COPYRIGHT (and COPYLEFT sometimes), or README. This will tell you whether you need to pay for the software package and/or restrictions on use and distribution of it. There seems to be a tendency to let public domain software slip through the cracks, but the license is there, just like any commercial product, and must be complied with. Legal penalties for infractions can be severe.

3.3 Installing a Software Package

Although this guidance obviously cannot be site-specific, there are some general things to look for when configuring, compiling and installing public domain software on your system.

3.3.1 Directory Structure

Set up a directory structure for building and installing public domain software. For instance, keep public domain packages in /opt/contrib, with the standard subdirectories like src (source files), bin (executables), man (manual pages) and lib (libraries). Most of the structure would be non-writable to all but systems administrators, except for the src directory, where users who are permitted to compile software can work. However, they will still need a systems administrator to install it into the proper locations once the software is built.

3.3.2 Precompiled vs. Source Code Distributions

Decide whether to get a precompiled program binary or the source code. The precompiled option means a quick installation, but relies on your system having things in standard places. If you want to evaluate the suitability of a package, an easy quick install of a precompiled program may be the way to go. Building the software from scratch is more time consuming. Using the source code means ensuring all the build options are configured properly, but this could also mean the software will fit more snugly on to your system. An advantage of source distributions is that they often contain some demonstration or template configuration files and test cases, which may not come in the precompiled distribution.

Sometimes the precompiled programs are bundled in with the source code distribution, so you get the best of both worlds.

README Files

Look for a README or INSTALL file for instructions. Most public domain packages will have a README or INSTALL or at least a file with ‘readme’ or ‘install’ somewhere in its name. Look to it for instructions for setting up the software package. If you get a precompiled distribution with no documentation, there may be a README for the software available in the same place you found the software.

Further details on 'Configure Scripts', 'Imake Files', 'Make Files', etc. can be found at
<http://www.environment.gov.au/epcg/erin/guidelines/technical/web_setup/sw_maint/sw_maint.html>

3.4 Upgrading

Finally, there is the decision of when to upgrade the software. There is a balance between staying with your existing, stable setup and moving to a new version with some extra features, with which you are not familiar. Major version releases and security patch releases are probably the most important to get. You don't need every minor patch to fix problems you haven't even encountered. When upgrading an important software package, you should set it up in a development area first and feel confident in the new version's performance and reliability. Even after the upgrade, it may be a good idea to have the old software on hand in case of a failure.

3.5 Documenting Your Actions

It is important to keep good system documentation. Software installation documentation is useful for ease of recovery in case of failure, as a guide for installing a new version of the software or the same version on an updated system. The facts remain after you have gone.

What your documentation should contain:

- The name and version of the software package.
- An indication of the type of license for the package, whether it is free use or requires a license fee.
- The home site for the software and the location from where you obtained the software. This will indicate where to check for new versions and software documentation. You may also want to record the date you last checked these locations.
- Downloading and unarchiving instructions—where to save the software distribution and where to unarchive it. Remember to put in warnings about checking where the distribution will land, so you don't accidentally blot out existing files.
- Build instructions—what changes you made to the distribution. Document the things you had to check, even if you didn't change them this time, as they may take on different values if your system changes.

- Installation instructions—say where the final files will be installed to run on your system. If file names change between versions, this will describe which files should be removed, prior to the installation of a new version. Otherwise, you could get residual files in your installation directories.
- Bug List—keep a list of the problems reported about running the software package at your site. When new versions are released or your system changes, you can check whether the problems have gone away.

It's probably a good idea to keep your documentation away from your source directories on the off chance that you do wipe out the source accidentally. Also, while you are diligently keeping all the license information in hard copy in a 'License Folder', you may want to keep the installation notes with its respective license in the same folder. You should also decide what format you want to keep your documentation in. Since much of the format will be consistent, automating certain parts of the document generation could be an option.

Maintaining Log Files

Logs files are useful for detecting unauthorised accesses, tracing transaction faults and for generating usage reports. They also chew up a lot of disk space, and you need to develop a strategy for archiving these files.

4. Data and Information Management Tools

There is a wide range of types of environmental data relevant to the conservation of biological diversity: text, tabular data (datasets), spatial data (GIS and remote sensing data), images, sound recordings, video clips, etc. Some require special-purpose software, and even hardware, for acquisition, management, analysis and dissemination. Others can be managed with a variety of technologies, the choice among them depending on the complexity and size of the dataset, extent and frequency of update, the ways in which it is analysed and used, and the skills and technologies available.

4.1 Text Processors

Substantial volumes of data and information are currently stored in text form. The easiest way to store and manage text is using word-processing packages. Where text is reasonably well structured, it may be more appropriate to consider spreadsheet or database management systems. However modern word processors are so sophisticated, particularly those that support macros (pre-stored sets of processing instructions that can be called up with a couple of keystrokes), that simple databases of up to a couple of thousand records can be established and maintained within them.

The more sophisticated word processors merge into desktop publishing systems, where pages of text, along with embedded illustrations, can be composed ready for printing.

4.2 Spreadsheets

Spreadsheets are commonly used for tabular numeric data, in particular financial data but, again, these are becoming so sophisticated that they can be used to manage tabular-oriented text in simple databases. These can be very useful with relatively simple databases up to a few thousand records.

4.3 Database Management Systems

Database management systems comprise a suite of programs that typically manage large structured sets of persistent data, offering pre-planned standard reports and ad hoc query facilities to multiple users. These two functions are usually built by separate processes: database design and applications development.

A database management system (DBMS) can be an extremely complex set of software programs that controls the organisation, storage and retrieval of data (fields, records and files) in a database. It also controls the security and integrity of the database. The DBMS accepts requests for data from an application program and instructs the operating system to transfer the appropriate data.

When a DBMS is used, information systems can be changed much more easily as the organisation's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Data security prevents unauthorised users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of the database, called subschemas. For example, an experts database can contain all the data about an individual expert, but one group of users may be authorised to view only their fields of expertise and an institutional mailing address, while others are allowed access to their personal telephone number or email address.

The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. It can also maintain an 'audit trail', so records that have been modified inappropriately or corrupted can be restored to an earlier version. The DBMS can keep duplicate records out of the database; for example no two individuals with the same code numbers (key fields) can be entered into the database.

Query languages and report writers allow users to interactively interrogate the database and analyse its data, or stream off a subset of the data for further processing.

An information system is made up of subjects (institutions, individuals, etc.) and activities (programmes, projects, etc.). Database design is the process of deciding how to organise these data into record types and how the record types will relate to each other. The DBMS should mirror the organisation's data structure and process transactions efficiently.

Organisations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. For example one system could manage a complete data set on threatened species while another serves up a subset of these data to the World Wide Web.

The three most common organisations are the hierarchical database, network database and relational database. A database management system may provide one, two or all three methods. Inverted lists and other methods are also used. The most suitable structure depends on the application and on the transaction rate and the number of inquiries that will be made.

Database machines are specially designed computers that hold the actual databases and run only the DBMS and related software. Connected to one or more mainframes via a high-speed channel, database machines are used in large-volume transaction-processing environments. Database machines have a large number of DBMS functions built into the hardware and also provide special techniques for accessing the disks containing the databases, such as using multiple processors concurrently for high-speed searches.

The world of information is made up of data, text, pictures and voice. Many DBMSs manage text as well as data, but very few manage both with equal proficiency. Throughout the 1990s, as storage capacities continue to increase, DBMSs are beginning to integrate all forms of information. Eventually, it will be common for a database to handle data, text, graphics, voice and video with the same ease as today's systems handle data.

Object-oriented Database

A system offering DBMS facilities in an object-oriented programming environment. Data are stored as objects and can be interpreted only using the methods specified by its class. The relationship between similar objects is preserved (inheritance) as are references between objects. Queries can be faster because joins are often not needed (as in a relational database). This is because an object can be retrieved directly without a search, by following its object identifier.

The same programming language can be used for both data definition and data manipulation. The full power of the database programming language's type system can be used to model data structures and the relationship between the different data items.

Multimedia applications are facilitated because the class methods associated with the data are responsible for its correct interpretation. Material such as video clips, photographs, maps or sounds can be held in the database just as readily as more traditional text or numeric data.

Object-oriented databases (OODB) typically provide better support for versioning. An object can be viewed as the set of all its versions. Also, object versions can be treated as full fledged objects. OODBs also provide systematic support for triggers and constraints which are the basis of active databases.

4.3.1 Database Design

This involves identifying the structure and functionality of the database. The required sources of data are made clear, and the integration and processing techniques needed to achieve the desired outputs are identified. The design process gives rise to a *functional specification,* which is independent of both hardware and software, and does not assume any particular method of physical data organisation (in practice, the technology available—which may be constrained by budgetary limitations—may influence the design of the database).

An important part of the design process is *data modelling.* This is the analysis of data objects and the identification of the relationships among these data objects. A common approach is to use *entity-relationship (E-R) diagrams.* Quite simply, an E-R diagram depicts the contents of a database: an *entity* (shown as a rectangle) is an object (or 'thing') about which data are collected; and a *relationship* (shown as a line) shows the connections between the entities. The nature of the relationships between the entities indicates the number of occurrences of one entity that may be associated with a single occurrence of the other.

Three types of relationship are possible: one-to-one, one-to-many and many-to-many. For example, the entity 'Protected Area' may contain data such as the protected area names, legal status, size and so on. There may be a relationship to a 'Country' entity, indicating that each protected area is located within one or more countries (i.e. a one-to-many relationship). Although protected areas normally fall within the borders of a single country, being aware that it is possible for them to straddle more than one country has important implications for design. Indeed, failing to establish the correct relationship at an early design phase could restrict the development of the application at a later date. It is vital to identify problems in the design phase, before large investments have been made in implementation.

In summary, the design process provides:

- a stable base from which to coordinate the development of the database, including the selection of appropriate equipment for implementation;
- a conceptual model which is free of implementation considerations, and which can be used as a point of reference when adding to or modifying the functionality of the database; and
- a baseline from which an optimum physical data organisation can be produced.

4.3.2 Applications Development

This involves creating a fully-functioning database using the data management software selected for implementation. Entities in the database design become *tables* in the database, and attributes become table *fields*. The way in which relationships between the entities are dealt with depends on which software is used; if it does not support some types of data relationship, then this has to be resolved by altering the database design.

Each field in the database is documented in terms of its purpose, data type, size and order in its corresponding table. When pooled across all the tables of the database, these definitions are known as the *data dictionary* of the database, and provide a description of its content, format and structure. The data dictionary itself can then form part of the metadata for the database (see *BCIS Handbook: Metadata*).

After the database tables have been created, they are *populated* with data. If the data are already computerised, this may be achieved by directly importing them into the database, plus associated re-structuring and formatting. If the data are only available in hard copy form, they will need to be entered manually into the database via the keyboard or, in the case of maps, images and structured text, via other input devices such as scanners, digitising tablets and related software.

Most data management software packages enable developers to customise data entry procedures, for example by enforcing certain formats and validating or correcting data items as they are entered. This concept can be extended to other procedures, such as the querying and reporting of data, and saving data to removable media (e.g. a floppy disk or CDs) for backup or delivery to users. The combination of data entry, querying and reporting features, security features and, of course, the underlying data tables, is known as a database *application*.

Database applications need not be created perfectly at the first attempt. Indeed, there is an advantage in developing *prototype* applications over a short time frame, and at low cost, in order to provide users with a means of refining their needs from the database. Prototyping is useful, perhaps essential, during the development of databases. The aim is to allow problems to be identified and corrected early on in the database development process, circumventing costly modifications at a later stage.

4.4 Geographic Information Systems[3]

A Geographic Information Sustem (GIS) is a computer system for capturing, storing, checking, integrating, manipulating, analysing and displaying data related to positions on the Earth's surface. Typically, a GIS is used for handling maps of one kind or another. These might be represented as several different layers, where each layer holds data about a particular kind of feature (e.g. roads or administrative boundaries). Each feature is linked to a position on the graphical image of a map. Layers of data are organised to be studied and to perform statistical analysis, examine the implications of alternative options and, more generically, provide information to support decision making.

4.4.1 Vector GIS

These are software and occasionally, though increasingly rarely, dedicated hardware systems for manipulating and displaying vector data. Some of the more sophisticated systems also manipulate raster data and have utilities for converting data from one form to the other. Such systems also incorporate sophisticated file management utilities.

Vector Data

An abstraction of the real world where positional data are represented in the form of co-ordinates. In vector data, the basic units of spatial information are points, lines and polygons. Each of these units is composed simply as a series of one or more co-ordinate points, for example, a line is a collection of related points, and a polygon is a collection of related lines. Vector data may or may not possess topological relationships. Typical vector data include administrative boundaries (polygons), road networks (lines) and sites where rare species have been recorded (points).

Topological Relationships

These indicate the relative location of geographic phenomena independent of their exact position. In digital data, topological relationships are usually expressed as relationships between nodes, links and polygons. This is useful in GIS because many spatial modelling operations can use

[3] Derived from an on-line dictionary of GIS terms provided by the Association for Geographic Information and the University Of Edinburgh Department of Geography <http://www.geo.ed.ac.uk/agidict/welcome.html>

topological information, for example, modelling water flow in a catchment. Water 'flows' only one way along connected line segments representing individual streams and rivers in the basin. From a series of field measurements that are stored as attributes of different line segments, a model can estimate total discharge from the entire catchment. Other models can use the same information to predict erosion rates in different parts of the catchment, sediment deposition in the downstream estuary, or flood risk in vulnerable areas.

4.4.2 Raster GIS

These are software systems for manipulating and displaying raster data. They tend to specialise in raster data and have few, if any, capabilities for handling vector data, other than converting any such data that arrives in a standard format.

Raster Data

An abstraction of the real world where spatial data are expressed as a matrix of cells or pixels, with spatial position implicit in the ordering of the pixels. With the raster data model, spatial data are not continuous but divided into discrete units. This makes raster data particularly suitable for certain types of spatial operation, for example overlays, area calculations, or simulation modelling, where the various attributes for each pixel can be readily manipulated because they are referenced to a common geographic base. Unlike vector data however, there are no implicit topological relationships. Remote-sensed data are largely stored and manipulated in raster form.

5. Analysis and Modelling Tools

There is a huge array of software systems and tools available. These range from general-purpose systems marketed by major software houses (e.g. RDBMS and GIS systems) to highly customised applications by individuals and research groups. The former are usually reliable and well supported, with regular update and maintenance programs, the latter can be idiosyncratic and unpredictable, and erratically supported, if at all.

Information on commercial 'off the shelf' systems can be obtained from many sources, so they will not be considered further here. On the other hand, attempting to cover the multitude of highly-specialised applications is well beyond the scope of this Handbook. The following is intended to provide pointers to some of the tools available. Further examples can be found in the Annex.

5.1 Software Directories

5.1.1 Software Tools for the Management and Visualisation of Biodiversity Data

This compilation by Richard Podolsky for The United Nations Development Programme was intended to assist biodiversity professionals accomplish several key tasks. First, the report would allow them to find useful software packages for the management and visualisation of their data. Second, it would allow them to communicate directly with publishers, authors and users of biodiversity software. Third, the report would enable them to use the Internet and World Wide Web as a resource for accessing software and technical information. Lastly, the report has information that will let biodiversity professionals participate in various internet discussion groups and upcoming conferences relevant to biodiversity <http://www.sdnp.undp.org/biod/bio.html>.

5.1.2 WWW Server for Ecological Modelling at the University of Kassel

This WWW-server provides access to available information about ecological modelling (simulation models, descriptions of these models, simulation-software, data sources and other information about modelling). It is also designed for modellers who want to make their models available. In addition, this WWW-Server integrates an interface to ECOBAS (Documentation of mathematical formulation of ecological processes) <http://dino.wiz.uni-kassel.de/ecobas.html>.

5.2 On-line Mapping and Modelling

5.2.1 Biodiversity Species Workshop

Biodiversity Species Workshop (BSW) allows researchers to perform a variety of functions on using biodiversity data. These include submitting their own latitude and longitude data from a web browser, conducting preliminary mapping and manipulation of that data, developing predictions of spatial distribution using that data, changing distributions with climate change, and visualising the outputs using GIS, animation, and virtual reality. BSW integrates data access, analysis and visualisation using CGI Perl scripts in a frame-based web application <http://biodi.sdsc.edu/bsw_home.html>.

5.2.2 ERIN Species Mapper

The ERIN Species Mapper is a facility to map and model the distributions of species on-line on the World Wide Web. Data are retrieved from a database containing records of the occurrence of plants and animals in Australia. These records have been obtained from Australian herbaria, museums and conservation agencies <http://www.environment.gov.au/search/mapper.html>.

6. Information Dissemination

6.1 World Wide Web

6.1.1 Finding WWW Software on the Internet

The sources below should be useful not only for WWW servers, but also for WWW clients and supporting multimedia software.

World Wide Web FAQ

This document has sections on where to find WWW related software, and how to use it. <http://sunsite.unc.edu/boutell/faq/wwwfaq.txt>. This document is available from many sites and in several languages. This FAQ consists of many files. It is available as an MSDOS .ZIP file, as a Unix compressed .tar file, and as a single, large text file. If you have trouble browsing HTML files offline under Windows, see the relevant FAQ entry.

Yahoo—Computers:Internet:World Wide Web:HTTP:Servers

Yahoo is a huge list of lists. This is its WWW server section. <http://dir.yahoo.com/Computers_and_Internet/Software/Internet/World_Wide_Web/Servers/>

NCSA Mosaic FAQ: Other Mosaic/WWW Software

Although a FAQ for the Mosaic WWW client, this should be of interest to people looking for WWW server software as well: <http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/faq-software.html>

W3 Search Engines

If you know, or think you know, what you are looking for, but don't have the locations, maybe this page of WWW search engines will be of use: <http://cuiwww.unige.ch/meta-index.html>

References for WWW Servers

This reference section contains home page locations dedicated to particular WWW servers. <http://www.environment.gov.au/epcg/erin/guidelines/technical/web_setup/www_servers/www_servers.html>

6.2 Compact Disc Read-Only Memory (CD-ROM)[4]

CD-ROM is a non-volatile optical data storage medium using the same physical format as audio compact discs. It is readable on computers with a CD-ROM drive.

CD-ROM is popular for distribution of large databases, software and especially multimedia applications. The maximum capacity is about 600 megabytes. A CD can store around 640 megabytes of data—about 12 billion bytes per pound weight.

CD-ROM drives are rated with a speed factor relative to music CDs (1x or 1-speed which gives a data transfer rate of 150 kilobytes per second). 20x was thought to be the maximum speed due to mechanical constraints but, in 1998, Samsung Electronics introduced the SCR-3230, a 32x CD-ROM drive that uses a ball bearing system to balance the spinning CD-ROM in the drive to reduce noise and caching to achieve speeds of up to x50 (December 1999).

There are several formats used for CD-ROM data, including Green Book CD-ROM, subsequently superseded by White Book CD-ROM and Yellow Book CD-ROM. ISO 9660 defines a standard file system.

---

[4] From "The Free On-line Dictionary of Computing, http://wombat.doc.ic.ac.uk/, Editor Denis Howe"

7. Acronyms and Technical Terms[5]

CGI
Common Gateway Interface, a standard for running external programs from a World Wide Web HTTP server. The CGI program can, for example, access information in a database and format the results as HTML. Perl is a common choice for writing CGI scripts.

client
A computer system or process that requests a service of another computer system or process (a "server"), using some kind of protocol, and accepts the server's responses. A client is part of a client-server software architecture. For example, a workstation requesting the contents of a file from a file server is a client of the file server.

client-server
A common form of distributed system in which software is split between server tasks and client tasks. A client sends requests to a server, according to some protocol, asking for information or action, and the server responds. This is analogous to a customer (client) who sends an order (request) on an order form to a supplier (server) who despatches the goods and an invoice (response). The order form and invoice are part of the "protocol" used to communicate in this case. There may be either one centralised server or several distributed ones. This model allows clients and servers to be placed independently on nodes in a network, possibly on different hardware and operating systems, as appropriate to their function, e.g. fast server/cheap client.

Compact Disc (CD)
A 120-mm disc comprising a polycarbonate substrate, a reflective metalised layer, and a protective lacquer coating. The physical format of CDs is described by the ISO9660 industry standard. CD-Recordable discs also have an organic dye data layer between the substrate and the metal reflective layer. They can store, on the same disc, still and/or moving images in monochrome and/or colour; stereo or two separate sound tracks integrated with and/or separate from the images; and digital program and information files. The same fabrication process is used to make both audio CDs and CD-ROMs for storing computer data, the only difference is in the device used to read the disk (the player or drive).

[5] Mostly derived from "The Free On-line Dictionary of Computing, http://wombat.doc.ic.ac.uk/, Editor Denis Howe"

CD-R
Compact Disc-Recordable. A 'write-once' version of CD-ROM. The term used to describe the technology of recordable CD as well as the equipment, software and media used to make recordable discs. The standard is described in the 'Orange Book'.

file server
see server

ISO 9660
The ISO standard defining a file system for CD-ROMs.

LAN
Local Area Network. A data communications network that is geographically limited (typically to a 1 km radius) allowing easy interconnection of terminals, microprocessors and computers within adjacent buildings. Ethernet and FDDI are examples of standard LANs.

object
In object-oriented programming, a unique instance of a data structure defined according to the template provided by its class. Each object has its own values for the variables belonging to its class and can respond to the messages (methods) defined by its class. An example of a class might be 'person', with an object being an individual's personal record. The attributes for that record (name, address, date of birth, etc.) would be those determined by its class.

Object-oriented programming
Writing programs in one of a class of programming languages and techniques based on the concept of an "object". Operations on the data can only be performed via defined methods, which are common to all objects that are instances of a particular "class". Thus the interface to objects is well defined, and allows the code implementing the methods to be changed so long as the interface remains the same. Each class is a separate module and has a position in a "class hierarchy". Methods or code in one class can be passed down the hierarchy to a subclass or inherited from a superclass. When a message is sent to an object, the method is looked up in the object's class, or if necessary in its superclass or higher, to find out how to perform that operation on the given object.

Orange Book
The specification for CD-R. Unrelatedly, also a standard from the US Government National Computer Security Council (an arm of the U.S. National Security Agency), "Trusted Computer System Evaluation

Criteria, DOD standard 5200.28-STD, December 1985" which defines criteria for trusted computer products.

OSI
Open Systems Interconnect. A model of network architecture and a suite of protocols (a protocol stack) to implement it, developed by ISO in 1978 as a framework for international standards in heterogeneous computer network architecture. The OSI architecture is split between seven layers, from lowest to highest: 1 physical layer, 2 data link layer, 3 network layer, 4 transport layer, 5 session layer, 6 presentation layer, 7 application layer. Each layer uses the layer immediately below it and provides a service to the layer above. In some implementations a layer may itself be composed of sub-layers. OSI is the umbrella name for a series of non-proprietary protocols and specifications, comprising, among others, the OSI Reference Model, ASN.1 (Abstract Syntax Notation 1), BER (Basic Encoding Rules), CMIP and CMIS (Common Management Information Protocol and Services), X.400 (Message Handling System, or MHS), X.500 (Directory Service), Z39.50 (search and retrieval protocol used by WAIS), and many others.

Perl
Practical Extraction and Report Language. An interpreted language developed by Larry Wall and distributed over Usenet. Perl is a general purpose language, often used for scanning text and printing formatted reports. Perl's speed and flexibility make it well suited for form processing and 'on-the-fly' page creation.

server
A program which provides some service to other (client) programs. The connection between client and server is normally by means of message passing, often over a network, and uses some protocol to encode the client's requests and the server's responses. The server may run continuously (as a daemon[6]), waiting for requests to arrive or it may be invoked by some higher level daemon which controls a number of specific servers. There are many servers associated with the Internet. Also a computer that provides some service for other computers connected to it via a network. The most common example is a file server which has a local disk and services requests from remote clients to read and write files on that disk.

---

[6] A permanently running program that lies dormant and is able to respond to specific conditions and requests.

SCSI
Small Computer System Interface. A processor-independent standard for system-level interfacing between a computer and intelligent devices including hard disks, floppy disks, CD-ROM, printers, scanners and many more. A range of different types of SCSI interfaces are available.

Unix
An interactive time-sharing operating system invented in 1969 by Ken Thompson. Dennis Ritchie, the inventor of C, is considered a co-author of the system. The turning point in Unix's history came when it was reimplemented almost entirely in C during 1972-1974, making it the first source-portable OS. Unix is a flexible and developer-friendly environment. By 1991, Unix had become the most widely used multi-user general-purpose operating system in the world. Unix is now offered by many manufacturers and is the subject of an international standardisation effort with the Unix trademark being owned by X/Open. Unix or Unix-like operating systems include OSF, Version 7, BSD, USG Unix, Xenix, Ultrix, Linux, and GNU.

WAIS
Wide Area Information Servers. A distributed information retrieval system. Clients are able to retrieve documents using keywords. The search returns a list of documents, ranked according to the frequency of occurrence of the keyword(s) used in the search.

WAN
Wide Area Network. A network, usually constructed with serial lines, extending over distances greater than one kilometre.

White book
A more open CD-ROM standard than Green Book. All films mastered on CD-ROM after March 1994 use White Book. Like Green Book, it is ISO 9660 compliant, uses mode 2 form 2 addressing and can only be played on a CD-ROM drive which is XA (Extended Architecture) compatible. White book CDs are labelled "Video CD".

Yellow Book
A CD-ROM format which is ISO 9660 compliant and uses mode 1 addressing. Discs of this type can be played on most drives and would be appropriate for most multimedia applications which have been developed for personal computers.

Annex

Management, Analysis and Modelling Tools[7]

There are many commercially available systems by major software manufacturers that can be adapted for use for biodiversity data. The following are examples of customised applications developed by individual agencies for their own use and for sale and distribution to others.

Name Type	WWW URL
Alice data management	http://dspace.dial.pipex.com/alice/
ANUCLIM model climate variables, bioclimatic parameters, and indices relating to crop growth	http://cres.anu.edu.au/software/anuclimtxt.html
BCD Biological and Conservation Data System	http://www.consci.tnc.org/src/bcdover.html
BG-BASE collections management	http://www.rbge.org.uk//BG-BASE/welcome.htm
BG-RECORDER plant records management for botanic gardens	http://www.rbgkew.org.uk/BGCI/database.htm
BIOCLIM model bioclimatic distributions	see ANUCLIM example: http://www.environment.gov.au/search/mapper.html
BioLink data and collections management and analysis	http://www.ento.csiro.au/biolink/biolink.html

(Continued)

[7] This list does not purport to be exhaustive and no warranty is given or implied as to the fitness of any of these tools for any particular application. Potential users must make their own assessments.

Name Type	WWW URL
Biota data and collections management	http://viceroy.eeb.uconn.edu/biota
BioTrack data management (uses Biota)	http://www.bio.mq.edu.au/kcbb/biotrack/default.html
BRAHMS Botanical Research And Herbarium Management System	http://www.camel.co.uk/brahms/
Carto Fauna-Flora mapping software to represent animals and/ or plant distributions	http://panoramix.umh.ac.be/zoologie/cff/cff_en.html
CASSIA Collections and Specimen System for Information and Analysis	http://www.nybg.org/bsci/cass/spec.html
CLIMEX predicting the potential distribution and relative abundance of species in relation to climate	http://www.ento.csiro.au/research/pestmgmt/climex/climex.htm
Condor planning tool that integrates biodiversity, social, and economic variables	http://www.conservation.org/SCIENCE/CPTC/INFOTOOL/Condor1.htm
DELTA DEscription Language for TAxonomy	http://www.keil.ukans.edu/delta/
DYMEX population modelling	http://www.ento.csiro.au/research/pestmgmt/dymex/dymexfr.htm
GARP designed for predicting the potential distribution of biological entities from raster based environmental and biological data	http://biodi.sdsc.edu/Doc/GARP/Manual/manual.html

(Continued)

Name Type	WWW URL
Linnaeus II biodiversity documentation and species identification	http://www.eti.uva.nl/Products/intro_linn.html
MEKA identification of biological specimens	http://www.mip.berkeley.edu/meka/meka.html
Orde management of data on the ecology and distribution of insects	http://www.cs.uu.nl/people/jeroen/orde/hl.html
PANDORA biodiversity research projects	http://www.rbge.org.uk/research/pandora.home
Platypus management of taxonomic, geographic, ecological and bibliographic information	http://www.ento.csiro.au/platypus/platypus.html
PRISMA publish databases, generate reports and training materials, and information about ecosystems	http://www.conservation.org/SCIENCE/CPTC/INFOTOOL/Prisma1.htm
SimCoast a fuzzy logic rule-based expert system for analysis of information collected on coastal transects	http://www.ccms.ac.uk/simcoast.htm
Species Analyst a software extension for ESRI's ArcView GIS software that provides an interface to species distribution prediction models	http://chipotle.nhm.ukans.edu/documentation/applications/SpeciesAnalyst/

(Continued)

Name Type	WWW URL
SysTax systematic botany and the administration of botanical gardens, herbaria and other collections	http://www.biologie.uni-ulm.de/systax/systax-e.html
WORLDMAP analysis and mapping	http://www.nhm.ac.uk/science/projects/worldmap/