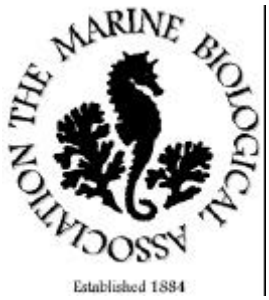

Marine Life Information Network (MarLIN)

Assessing seabed species and ecosystems sensitivities. Software development scoping study.

Daniel Lear

February 1999

Reference:

Lear, D., 1999. *Assessing seabed species and ecosystems sensitivities. Software development scoping study.* Report to the Department of the Environment Transport and the Regions from the Marine Life Information Network (*MarLIN*). Plymouth: Marine Biological Association of the UK. (*MarLIN* Report No.2.)

Assessing seabed species and ecosystems sensitivities.
Software development scoping study.

Contents:

Marine Life Information Network (*MarLIN*)

Assessing seabed species and ecosystems sensitivities. Software development scoping study.

Daniel Lear

1 Introduction

'Identifying species and ecosystems sensitivities' is a project funded by the Department of Environment, Transport and the Regions (DETR) and contributes to the biology and sensitivity key information sub-programme of the Marine Life Information Network (*MarLIN*). Objective 2 of the DETR project is:

> *Develop a user-friendly computer-based system that will allow the information thus gathered to be interpreted and used by decision-makers applying the ecosystem approach to environmental management.*

This scoping study sets out to outline the requirements of software developed to achieve the above objective and approaches that may be developed. The advantages and disadvantages of each system are considered and how decisions about the systems development have been reached are explained.

2 Background

Current information technology provides us with the means for global dissemination of scientific information for practical applications, peer review and educational purposes. Vast resources are now available through the World Wide Web bringing collaborative researchers and the scientific community closer together.

The 'thin client' paradigm has become the driving force of Web technology with information being pulled from a central server rather than being stored on the users machine. Web browsers use a technology called 'pull' since the information is pulled from web servers through links. The 'push' technology (such as 'cookies') allows vendors to push pre-selected information down to users desktops. In 'pull' technology, all the processing of information, running of executable files and processing of scripts occurs on the server. This saves resources on the users machine and greatly improves speed and efficiency in information manipulation and retrieval. 'Thin' clients

also permit a much higher degree of interoperability, meaning the same processes can be run on a variety of platforms and operating systems (LaOr 1997).

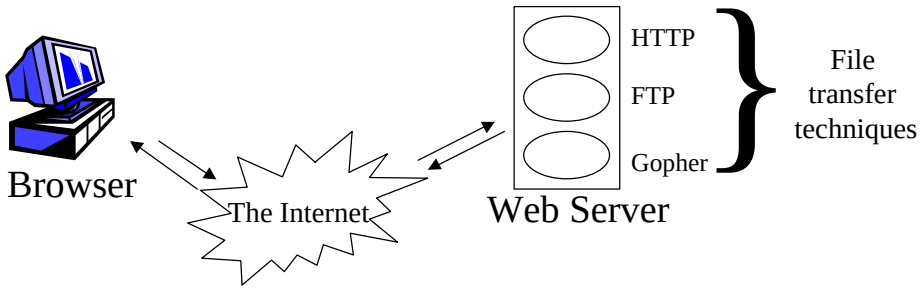

Figure 1.How a web browser communicates with the server. The browser is the software used to access the Internet; Http (Hypertext transfer protocol), Ftp (File transfer protocol) and Gopher are all methods for the movement of information over the Internet.

3 User Requirements

All information-system development and purchases involve the user requirements being identified. These user requirements guide the development of the data entry and access fields and how information is presented. The main guiding principles of the *MarLIN* software development are defined as:

- information must be distributable or viewable by all parties;
- software must be able to link with existing systems or those currently in development or planned for the future;
- data and information needs to be able to link with an interactive map or Geographical Information System (GIS), for spatial and/or temporal data representation, and
- the system needs to be useable by non-specialists who are unfamiliar with specific terminology.

Based on these criteria and to make the interface as intuitive and as user-friendly as possible, it was concluded that a Web-based format would be most applicable. This would make all of the information accessible through the standard Internet browsers, such as Microsoft Explorer or Netscape Navigator. Utilising several scripting and programming techniques, data can be linked into web pages from a variety of other disparate software packages, thus preventing users being dependent on a specific software package.

The Web approach facilitates a simple 'point and click' method of navigating through the data and linking to external sources of information. This however also presents a potential problem. The various Web browsers available will interpret raw HTML (HyperText Mark-up Language, the standard language that simple Web pages are authored in) in a variety of different ways, some subtly, some vastly different from the original format. This can result in greatly differing appearances depending upon which browser is being used. This is a vital consideration in the development of any web page and one that needs to be accounted for very early in development.

4 The database

4.1 Software

In order to store and interrogate the large amounts of data that are to be included within *MarLIN's* activities it is necessary to utilise a database. Databases provide an excellent information storage and retrieval system for this type of data.

Several criteria were necessary when choosing which product would be utilised. The software must:

- be 'relational' in its structure (i.e. links between fields, dynamically updateable data);
- be accessible through the Internet;
- permit data entry and manipulation through the Internet and through all forms of browsers;
- allow complex query formation, based on existing data;
- be compatible with other data systems currently in development, such as those within Joint Nature Conservation Committee (JNCC) and the nature conservation agencies;
- provide sufficient security for data held within it;
- allow data to be readily imported and exported from it;
- be flexible in its design, and
- have sufficient capacity for all the data that will be accumulated.

Based on these factors, Microsoft Access has been chosen as the database that will provide the backbone of the system. It is widely used, and satisfies all the criteria specified above and in

addition is extensively programmable, allowing greater customisation in tailoring the package exactly to the needs of the sub-programme.

4.2 Information fields

The sorts of information required to describe the biology of species and biotopes and to identify sensitivity and recoverability have been bought together for the *MarLIN* programme by Dr Keith Hiscock and have been developed and promulgated in various fora. Development has especially occurred as a result of discussion at the recording workshop held in February 1998 in Newcastle (Foster-Smith 1998), a meeting of the Project Management Group in November 1998 and a workshop held in Bangor in January 1999. In addition these fields have been used to produce reviews of key information as a background to Species Action Plans (Part of the UK Biodiversity Action Plans) and for the OSPAR IMPACT meeting, held in September 1998.

Within these fields it was necessary to develop indices for assessing sensitivity and recoverability. Workshops held under the auspices of the ICES Benthos Working Group and reviews for the OSPAR IMPACT as well as studies commissioned by the nature conservation agencies contributed to the development of the indices (Hiscock, Jackson & Lear 1999; Holt 1995,1997; Anderson & Moore 1997; Cooke & McMath 1998).

The structure of the database was designed to be extremely flexible, taking advantage of the inherent features within MS-Access. Great care is needed in not constraining the user with the choices presented, whilst still enforcing some degree of data integrity. This can be achieved by means of drop-down boxes, and also by creating validation rules that are applied to fields to restrict the format of what can be entered.

4.3 The user interface

4.3.1 Introduction

To dynamically distribute the information held in the database via the Web interface requires complex programming and extensive trialling of the various approaches. Unfortunately many new technologies that could be used for the dissemination of the data are not fully supported by all Web browsers and so are excluded from consideration.

It is also essential that the data held within the database is secure and the web interface cannot be used to edit or delete information without authorisation. All information displayed will be refereed prior to being made globally available.

The user interface must:

- be clear, concise and accessible by all;
- allow access to glossaries to explain terminology and concepts;
- permit protection of the data, to prevent unauthorised data entry and/or deletions;
- link with MS-Access database dynamically through the Web server, and
- enable complex querying and manipulation of the data remotely.

It is vital to consider the above factors when choosing how the software is to be developed. Two approaches are detailed below.

- Common Gateway Interface (CGI)
- Active Server Pages (ASP)

4.3.2 Common Gateway Interface

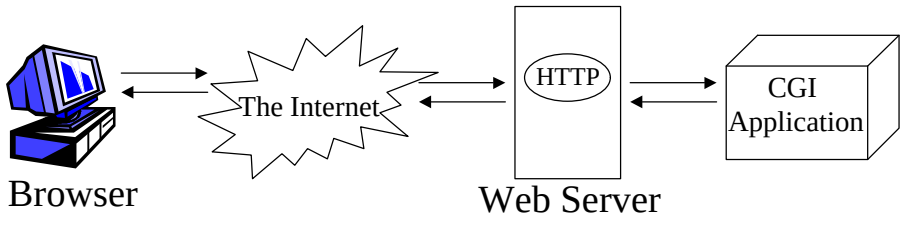

Figure 2. How CGI interfaces with the users browser.

The Common Gateway Interface (CGI) is a standard for interfacing external applications with Web servers. CGI was originally developed using UNIX shell scripts and the Perl language. The CGI program which is run by the users browser executes on the server in real-time and is therefore capable of outputting dynamic information. This information is passed from the server to the CGI application, and *vice versa*, using environmental variables.

CGI scripts can be written in a wide variety of programming languages including C/C++, Visual Basic, Fortran, Perl, TCL, AppleScript and within any UNIX shell. This provides a great deal of inherent flexibility for script authors.

Initial attempts to run CGI in the Windows environment were based around the DOS system, this was due to Windows having no basic command interpreter. The main disadvantage with this approach was the necessity to begin a new DOS session for each CGI request made to the server, thus limiting the number of CGI processes that could be run at the same time. In addition, multiple DOS sessions can potentially result in memory leaks, eat up the servers memory and require the servers operator to reboot the machine to restore the memory.

The Windows Common Gateway Interface (Windows CGI) was developed to overcome the limitations of DOS CGI. As previously mentioned, Windows has no native command interpreter (Windows NT is an exception). Therefore, the Windows CGI application needs to be a Windows executable, that is a pre-compiled file. In order to minimize programming efforts and keep the interface simple, data from server to client, and vice versa, is passed via input and output files instead of environment variables.

By utilising files that form part of the application programming interface (API) programmers are permitted to read the CGI input files quickly and reliably from within a programming environment such as Visual Basic or C/C++. This is possible as the CGI input files share the same format as the Window initialisation (.INI) files (LaOr 1997).

The Windows CGI standard includes three distinct ways for the sending of information from the server to the application, and one standard method for receiving information, each used in different circumstances. This allows a great deal of flexibility in the manner in which data is manipulated and distributed.

4.3.3 Active Server Pages

Active Server Pages were developed by Microsoft as a language independent framework to ensure efficient coding of scripts that are executed by a Web server in response to requests made by the user. It enables mixing of HTML and ODBC (Object Database Connectivity) database reading and writing whilst avoiding the need for tools like CGI, JavaScript, Perl and ActiveX. The transfer of data from the database via ODBC is controlled by a Relational Database

Management System (RDBMS). It is possible to create rudimentary ASP Web pages directly from Microsoft Access with simple automated steps. However, to gain the most from the technology some knowledge of the underlying principles is essential.

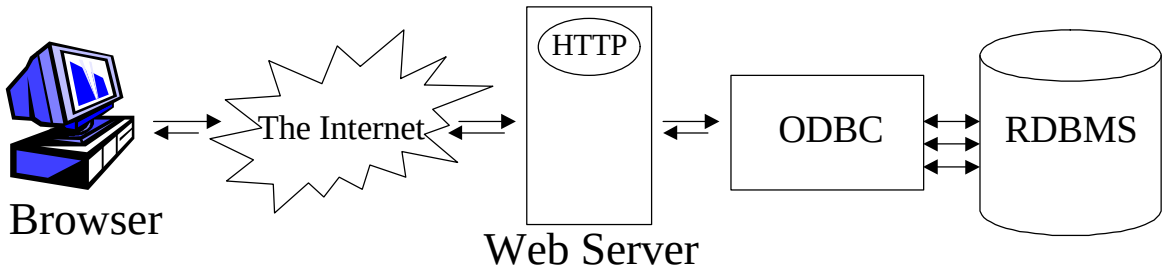

Figure 3. How ODBC interfaces with the database and the users browser.

With ASP, all scripting code runs on server, and the browser just receives resultant HTML so the receiving browser needs no special capabilities and all pages are displayed in a standardised format (Asp-zone.com 1999).

ASP is now widely used and its uses are displayed on many sites throughout the Web. However, despite the fact that ASP is so widely used and has a wide range of information published about it, no standards body has approved it.

When the user interrogates the system to retrieve information, a complex chain of events occurs. The browser is responsible for submitting the query and displaying the results of that query. Then the Web server accepts the query from the browser, creates a connection to the database, queries the database, formats the results into HTML and delivers the resultant HTML to the requesting browser. The database server is responsible for accepting requests from the Web server and delivering them back to the Web server. In this approach it is the Web server that acts as the client to the database server and no connection is directly made between the browser and the database server. This is important when considering security related 'permissions' to the database server (Johnson 1997).

ASP has three main advantages:

Performance	The Active Server Pages host is not re-launched with every script access and is therefore much more efficient. Unlike CGI scripts, which are run as executables in

	their own processes. This results in a new process being created each time a script is requested from the server. The server must then run the script and kill the (just created) process. This is inefficient and can have a severe impact on the performance of the Web server.
Session management	ASP provide a built-in session management functionality that allows developers to persist data (i.e. to keep track of information even though a user might switch pages) for the duration of a connection to the database.
Easy integration of COM components	ASP is designed to rely heavily on COM (Common Object Model) components for its wide usability. As a result, it is very easy to use any COM component from within an ASP script. This facilitates database (or most other software) integration.

As Microsoft developed ASP, their capacity to mesh seamlessly with the Access database is unparalleled. However, the level of user control is reduced to some extent when compared with established technologies such as CGI.

4.3.4 Conclusion

The two techniques described above both have a part to play in the development of the *MarLIN* data system. As they are not mutually exclusive it would be of benefit to include them both.

5 Data security

As with any computer system, it is vital to ensure the security of both the information held and the hardware itself. The database and associated Web site will be backed up to a digital tape on a weekly basis and stored in a fireproof safe. The site itself will be protected through the inherent security systems within the Windows NT operating system. This will prevent and avoid malicious or accidental deletion or alteration of the data held by *MarLIN*. The data can be secured through simple user recognition or by a more complex password scheme, both are easily implemented and integrated into the system.

6 Validation and testing of the data-systems

Once the database is constructed, including all the key fields as recommended by the Project Management Group, it is vital to ensure the system is:

- Intuitive and simple to use.
- Robust in its design.
- Easily distinguishable from other products.
- Attractive in appearance and transition through the data.

Trialling will be undertaken in the first instance by the actions of Angus Jackson, the Data Researcher for the sub-programme. His entry of data into the system should highlight any shortcomings in the system design and in data interrogation and representation. Once beyond this stage the system will be available on-line for peer review by the Project Management Group for further comments on its content and appearance. Finally, providing they are satisfactory, the pages will be viewable by all, through the World Wide Web. Other aspects of data acquisition specifically concerned with data quality, access permissions and scoring systems are being addressed elsewhere in the project.

7 Expected timetable

The development of the database has already commenced, with all the key fields for species in place. Establishment of data fields for habitats and biotopes will commence in April. The database is currently being tested by the Data Researcher. The next stage is to move the data entry point from being through the MS-Access forms over to a web-interface. This stage will be more in line with the final product, and permit further assessment of the fields and user-interface. The development of a web interface will involve a considerable amount of programming, testing and validation. It is anticipated that a functional web-interface will be available through a limited Web connection by mid-June, 1999. This timetable will not prevent the database being populated, but will ensure the stability of the system once available globally.

8 Acknowledgements

Keith Hiscock specified report requirements and, with Angus Jackson and Emily Wilson, reviewed drafts. Members of the Project Management Group and participants at the Bangor workshop in particular provided comments on initial proposals.

9 References

Anderson, S. & Moore, J. (1997). *Guidance on assessment of seabed wildlife sensitivity for marine oil and gas exploration.* A report to JNCC from OPRU, Neyland, UK. Report No. OPRU/18/96.

Asp-zone.com. (1999). *Active Server Pages Frequently Asked Questions*. http://www.asp-zone.com/aspfaq.htm

Cooke, A. & McMath, M. (1998). *SENSMAP: Development of a protocol for assessing and mapping the sensitivity of marine species and benthos to maritime activities*. Bangor, Countryside Council for Wales (Marine Report No. 98/6/1.)

Foster-Smith, J. (1998). *Proceedings of the marine species recording workshop. 29th & 30th January 1998*. Peterborough, Joint Nature Conservation Committee. JNCC Report, No. 280.

Hiscock, K., Jackson, A & Lear, D. (1999). *Assessing seabed species and ecosystem sensitivities. Existing approaches and development.* Report to the Department of the Environment Transport and the Regions from the Marine Life Information Network (*MarLIN*). Plymouth, Marine Biological Association of the UK. (*MarLIN* report No. 1.)

Holt, T.J., Jones, D. R., Hawkins, S.J. & Hartnoll, R.G. (1995). *The sensitivity of marine communities to man-induced change*. Bangor, Countryside Council for Wales (Report No. 65.)

Holt, T.J., Jones, D. R., Hawkins, S.J. & Hartnoll, R.G. (1997). *The sensitivity of marine communities to man-induced change*. Nature Conservation and the Irish Sea seminar. 6th February 1997. Liverpool, Irish Sea Forum (Seminar report no. 15.)

Johnson, S. (1997). *Using Active Server Pages*. Que Corporation, Indianapolis.

LaOr, O. (1997). *CGI Programming with Visual Basic 5*. McGraw-Hill, New York.

Appendix 1: Glossary of Acronyms

API	Application Program Interface. An interface between the operating system and application programs, which includes the way the application programs communicate with the operating system, and the services the operating system makes available to the programs.
ASP	Active Server Pages. Active Server Pages is a compile-free application environment for Microsoft's Web Server. ASP scripts can be embedded in an HTML document and are parsed before being sent to the client web browser. Active Server Pages provide a method to create dynamic or database driven web pages. Active Server Pages have native support for both VBScript and Jscript and can also embed ActiveX server components.
CGI	Common Gateway Interface. A way of interfacing computer programs with HTTP or WWW servers, so that a server can offer interactive sites instead of just static text and images.
COM	Common Object Model. Microsoft's standard for object interaction. COM provides a standard mechanism by which objects and components can communicate, independent of the language in which the components were created.
DETR	Department of the Environment Transport and the Regions.
DOS	Disk Operating System. More computers worldwide have DOS than any other operating system.
FTP	File Transfer Protocol. A client/server protocol for exchanging files with a host computer.
GIS	Geographical Information System.
HTTP	HyperText Transfer Protocol. The protocol most often used to transfer information from World Wide Web servers to browsers. Also called Hypertext Transport Protocol
ICES	International Council for the Exploration of the Seas.
JNCC	Joint Nature Conservation Committee.
ODBC	Open Database Connectivity. An interface that makes it possible to access different database systems with a common language.
OSPAR	The Convention for the Protection of the Marine Environment of the North-East Atlantic.
RDBMS	Relational Data Base Management System. A complex set of programs that control the organization, storage and retrieval of data for many users in a relational

database; extensively used in business environments. Data is organized in fields, records and files. A database management system must also control the security of the database

UNIX A multi-user, multi-tasking operating system. It was originally designed for minicomputers, then revised for use on mainframes and personal computers. There are now many versions of UNIX which can be used on many different platforms. UNIX is written in the C programming language.

WWW World Wide Web. A hypermedia-based system for browsing Internet sites. It is named the Web because it is made of many sites linked together; users can travel from one site to another by clicking on hyperlinks. Text, graphics, sound, and video can all be accessed with browsers like Mosaic, Netscape, or Internet Explorer.