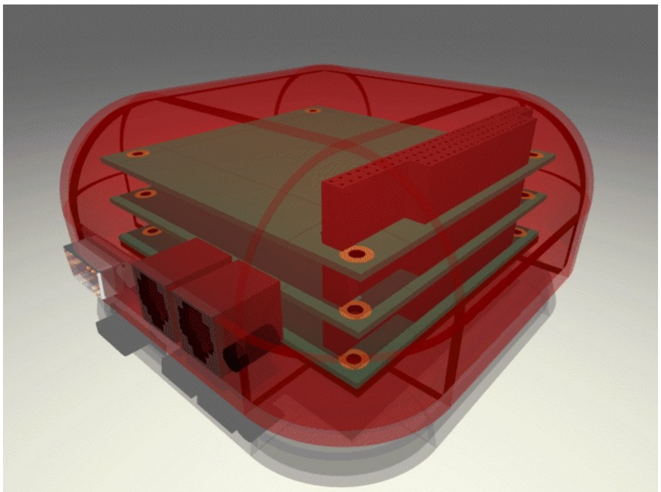
Low Cost Modem with Integrated Intelligence

3H Design Project



Author:	James Inge
Supervisor:	Adrian Vinsome
Date:	17 March 2000

University of Durham
School of Engineering

Executive Summary

This report describes the design of the **DataBug** - a **Low Cost Modem with Integrated Intelligence**. This device will allow remote reconfiguration of serially controlled equipment. It will reduce the time taken for reconfiguration and remove the expense of a service engineer visiting remote locations.

The DataBug's main selling point will be its **ease of use**. It will be capable of being installed, configured and used by untrained personnel. A **low purchase price** combined with **time and cost savings** will make the DataBug more attractive than competing systems.

Contents

Diagrams

Nomenclature

BABT	British Approvals Board for Telecommunications
CAD/CAM	Computer Aided Design/Computer Aided Manufacture
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DCE	Data Communication Equipment, e.g. a modem
DSP	Digital Signal Processing
DTE	Data Terminal Equipment, e.g. a console
EMC	ElectroMagnetic Compatibility
GUI	Graphical User Interface
HTML	HyperText Markup Language
I/O	Input/Output
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
modem	MOdulator/DEModulator, used to send data over an audio link.
MNP	Miracom Networking Protocol
OSI	Open Systems Integration, an ISO networking model.
PC	Personal Computer
PCB	Printed Circuit Board
PSTN	Public Switched Telephone Network
RTTE	Radio and Telecommunications Terminal Equipment
TAPI	Telephone Applications Programming Interface
TLA	Three Letter Acronym
UART	Universal Asynchronous Receive/Transmit, a type of serial communications interface chip.
USART	Universal Synchronous/Asynchronous Receive/Transmit, as above but more flexible.
USB	Universal Serial Bus
xDSL	Any of various Digital Subscriber Line technologies.
XML	eXtensible Markup Language

Introduction

In many different areas of industry equipment must be left running unattended, often in a location geographically remote from the operators. Examples of such equipment include pumping stations for oil, water, gas, or the chemical industry; data communications equipment such as network routers; burglar alarms or building management systems.

Many of these devices are fitted with serial ports to allow external control. Normally this will be via a computer. When the device to be reconfigured is in a remote location, operators are currently faced with two alternatives:

1. Send a technician with a laptop to adjust the equipment.
2. Leave a general purpose PC and modem at the site, to allow dial-up access to the equipment.

The first case is expensive in terms of time, travel expenses and wages. The second is costly due to capital expenditure, space consumed at the installation, and power required to run the system over a long period of time.

The DataBug is intended to overcome these problems. It will…

- Remove the need for post-installation technician visits.
- Mimic the functionality of a full modem-plus-computer installation.
- Be cheap enough to be deployed on a large scale.
- Be easier and quicker to use than any existing system.

Although the DataBug could be used for telemetry, it should be pointed out that it is intended primarily for use as a reconfiguration link. This role implies infrequent use and low volumes of transmitted data. Hence issues such as connection set-up time and maximum throughput rate are not of great importance in the design.

Design

Cost

A major remit of the DataBug is to be low cost. This has correspondingly influenced the design process. Savings will initially be achieved by reducing the number and cost of the components. Where possible, Commercial Off-The-Shelf (COTS) parts will be brought in, rather than custom developed solutions. This will reduce the lead time in the DataBug's manufacture, and allow it to be updated to track future technology developments with little extra design cost.

Care will be taken to design the product for manufacture, to reduce the time and costs of construction and assembly. This will involve reducing the number of screw fasteners, flying leads and other components that consume assembly time.

These measures can only reduce costs to a point, and in the technology marketplace it is not wise to compete on grounds of absolute cost alone. It is probable that someone else will be able to build the same device cheaper elsewhere.

Further reductions in the cost of the DataBug will therefore be achieved by increasing its value. Value can be added in several ways:

- By reducing installation time compared with other products.
- By reducing the learning time for both configuration and use.
- By making the DataBug easier and quicker to use competing solutions.

The above discussion leads to the conclusion that the design of the DataBug should be kept as simple as possible. This will reduce the costs associated with design, testing and manufacture. Simplicity will also make the device easier to install and quicker to understand, adding value. The DataBug will also add value to other products, by increasing the range of locations in which they can be used.

Concepts

Initial brainstorming revealed that there were two main parts to the design: connecting to the communications network, and providing control logic for the system. In the scenario we wish to replace, these parts equate to a modem and a PC respectively.

These two units could be combined at various levels of integration. A modem and control logic system could be constructed from discrete components. This would be very time consuming in the design phase, costly to construct and test, but not require any specialist equipment or techniques.

It might alternatively be feasible to construct a system around a microcontroller such as the Intel 8051 or equivalent, which incorporates a built in UART. This would mean that PCB design was reduced to building interfaces between the phone line, modem chip, serial port, microcontroller and memory. Developing this idea, a digital signal processing (DSP) chip could combine the majority of the necessary components.

The above ideas improve the cost over discrete components but are relatively inflexible and would still require extensive testing. Instead, the modem and processing elements of the device could be bought in as COTS modules. Electronic design would then be reduced to the "glue" necessary to ensure a working interface between modules. This approach also simplifies testing, by physically dividing the device into sections, which can be tried separately.

Proposal

After preliminary research it was decided to use the third of the above concepts: a Modular design. This gave several advantages:

The project's research stage would largely be limited to determining the hardware functions necessary for the device. Research into how to actually implement those functions would have already been done elsewhere. Using off-the-shelf parts would save us from re-inventing the wheel and allow us to concentrate on making our product competitive.

By using existing modules, the design emphasis of the project is shifts from electronics to software, packaging and the interfaces between the system's components.

Research had revealed a number of manufacturers producing suitable components, often in the PC/104[1] format. This standard defines a form factor and electrical specification for miniaturised PC and PC-AT compatible bus cards. PC/104 appeared ideal for our needs, since it was designed with the aim of producing small, low power, embedded applications, while maintaining compatibility with existing hardware and programming tools.

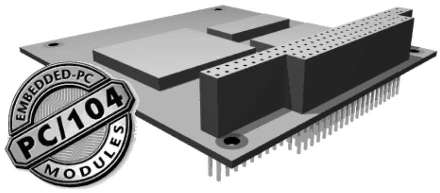


Figure 1. PC/104 form factor card.

Detail Design

At an early stage in the design process, it was decided to set out requirements for the DataBug and identify which standards it would meet. This was done for several reasons.

- The DataBug must comply with all the various EC product directives in order to be legal for sale. The easiest and cheapest way of accomplishing this is to consider them during the early stages of design.
- For a modular design to work, all its components must interface properly with each other. This means that they should all conform to the relevant standards, where they exist.
- Identifying specific requirements allows the design to be conducted with well-defined aims, and permits a proper evaluation of the solution.

Table 1. Requirements for DataBug.

Requirement	Relevant Standards
Manufacture All products must be sourced from quality assured suppliers. The number of connectors and fasteners used is to be minimal. Final assembly must take no longer than two minutes by hand. Standard form factors must be used throughout.	ISO 9001[2]
Installation Installation must require no special training. Installation must take less than five minutes. No tools are to be required for installation.	
Environment and Packaging The device should be fully operable in the temperature range 0º to 60ºC, ideally -40º to 85ºC. Any cooling necessary should be by air circulation, ideally passively. The packaging should be protected to at least IP 41. Ideally protection will be to IP 52. The device should be suitable for use in a Zone 2 hazardous area. The device should be packaged in a standard sized case.	EN 60529/(Cor:1993)[3]
Electrical Performance The device must conform to the EMC Directive. The device must conform to the Low Voltage Directive. The device must conform to the RTTE Directive. The device must conform to the BABT and Oftel guidelines for connection to telecommunications networks. If international versions are produced, they must conform to the equivalent regulations of the target states. Connections to the telephone network must use the proper connectors. Other appropriate national or international standards should be followed where possible.	89/336/EEC[4] 73/23/EEC[5] 99/5/EC[6] BS 6320:1992[7], BS 6328-1:1985[8], BS EN 41003:1999[9], BABT Regulations[10] BS 6312-2:1997[11]
Signalling RS-232-D communications should be supported. Ideally the device will support, or be extensible to support, other serial standards. The modem should comply with the ITU recommendations for signalling at up to 9600 BPS. This could ideally be extended to 14400 BPS or 33600 BPS. Error control protocols MNP 1 to MNP 4 must be supported. Ideally V.42 error correction may also be supported. The modem must negotiate the most error-resilient connection available. In order of preference this is V.42 then MNP 4 through MNP 1, with no error control protocol as a last resort. The modem might support V42.bis and/or MNP 5 data compression. V.42bis is preferable.	V.21[12], V.22[13], V.22bis[14], V32[15], (V.32bis[16], V.34[17]) MNP 1 – 4, V.42[18] V.42bis[19], MNP 5
Software Operation Learning time for the user interface should be less than ten minutes for basic use or one hour including set-up and configuration. The modem should be able to transmit character streams of data. User authentication should be performed. There must be a minimum of 10^9 combinations for any password used. Software should be designed in a portable, modular, hardware-independent manner as far as possible.	BS EN ISO 9000-3[20]

Software and hardware were designed using a top-down hierarchical approach. Higher levels were actually realised while lower, more hardware specific layers, were deemed unnecessary for this stage of design. The design can be considered using the OSI 7-Layer Model:

Table 2. 7-Layer model of DataBug System

Layer	Client Machine	Server Machine	DataBug	Equipment
1. Application	The front-end to the DataBug will run on a user's local equipment, within a web browser. Active parts of the interface could be build using technologies such as Java, JavaScript or VBScript.			Hardware dependent
2. Presentation	CSS[21] Stylesheets allow the interface to be correctly formatted for different devices, for example GUI's for PCs or workstations; or audio or reduced graphical interfaces for mobile phones.	XML[22] Command Sets describing equipment control are merged into the user interface. This is supplied to the user as an HTML[23] structured document.		Hardware dependent
3. Session	User initiates a session by choosing a DataBug to call.	Confirms the client is authorised to access the system, then dials up and logs in to DataBug.	Confirms valid user ID and password. Determines access level.	Hardware dependent
4. Transport	Encrypts commands, if necessary.	Modem adds error correction.	Modem provides error correction. Software decrypts data stream.	Hardware dependent
5. Network	Adds port selection data.	Sets up PSTN connection to DataBug.	Sends data to correct port.	Hardware dependent
6. Link	Client dependent	Server transmits data as 8 data bits, no parity, 1 stop bit (8N1).	Converts from 8N1 to equipment data format.	Hardware dependent
7. Physical	Client dependent	Server modem injects data onto PSTN.	Acts as a selective bridge between the PSTN and RS-232.	RS-232

Electronic Design

The DataBug's hardware forms a link at the Physical layer of the network (see Table 2). The first part of the link is the connection between the telephone line and the DataBug.

In order to be allowed to connect to a phone line various standards must be met, as shown in Table 1. By using a pre-built modem module, we are saved from designing the necessary features ourselves. Standard modems also provide various higher level functions, such as error correction, data compression and auto-negotiation of transmission speed. The modem would connect directly to the DataBug CPU module, and be controlled via the PC/104 bus. Control of the link would be via the standard Hayes "AT" Command Set.

Figure 2. PC/104 modem board.

The second stage in the DataBug link is the serial data transfer to the connected piece of equipment. The format of this RS-232 data is dependent on the particular hardware. Serial links may have just three wires, for transmit, receive and ground, or may include further handshaking signal pairs (see Table 3). Beyond this, the connectors may be 9- or 25-pin, male or female. The information transmitted can vary in baud rate and number of data, parity and stop bits. This standardisation problem is discussed in greater depth in Reference ([24]).

Table 3. RS-232 Pin Assignments

Name	Pin Number		Function
	9 pin	25 pin	
TD	3	2	Transmit data
RD	2	3	Receive data
RTS	7	4	Ready to send
CTS	8	5	Clear to send
DTR	4	20	Data terminal ready
DSR	6	6	Data set ready
DCD	1	8	Data carrier detect
RI	9	22	Ring indicator
FG		1	Frame ground
SG	5	7	Chassis ground

The DataBug should be able to cope with all these combinations. It will be fitted with a 9-pin connector, since this is physically smaller. It will be male, the style normally fitted to DTE devices such as computers. Adapters are readily available to switch from this format to 25-pin, as are "gender-benders" to ensure proper mating.

To cope with different data rates and bit formats, a programmable Universal Asynchronous Receive/Transmit module would be used, e.g. the National Semiconductor 16550AN. This module would deal with synchronising to a bit-stream, and properly formatting the data. An RS-232 line driver such as the Maxim MAX232 would be required to generate the required signal voltages.

These modules can be found as standard on PC/104 CPU boards, and can be controlled using standard PC device drivers. In considering the design, it was decided that another communications device should be added to the CPU board - a network interface.

Figure 3. PC/104 CPU board.

With a single serial port, the DataBug could normally only control one device. The equipment would need to be relatively close to avoid transmission errors due to the cable length. Increasing the number of ports would increase the required size of the DataBug's case, and the possibility of cross-talk between the cables.

These problems could be removed by using a separate network module consisting of an Ethernet access controller, a UART, line drivers and an array of eight ports. In the simplest case, it could be used with a straight-through connection to the DataBug. This would allow signals for eight devices to be multiplexed down one cable, and act as an intelligent extension lead.

If a live network were used, the DataBug could control devices in other parts of a building. One DataBug could control an almost unlimited number of serial devices. It could also command devices fitted with an Ethernet remote control port, rather than a serial interface. These would be linked directly to the network.

A suitable PC/104 CPU board has been identified: the AT/3i from EuroTech S.p.A (www.eurotech.it), which includes onboard Ethernet. The modem function would be provided by the Minimodem OEM #8410 from Aprotek (www.aprotek.com).

A third, in-house PC/104 board would also be required: the I/O board. This is a very simple board, used to hold the various connectors in the correct places, and route signals from them to the PC/104 bus. Our design would use 8-bit PC/104 cards, with the 16-bit socket used for board to board connections.

The I/O board would be fitted with a 12V power socket, two RJ-45 connectors (phone & network), and the 9-way RS-232-D serial connector.

Software

The first software component is the DataBug's embedded software. It performs the following functions:

- User authentication, via a password system.
- Configuration of serial ports for the connected devices.
- Transfer of data to and from connected equipment.

Configuration and control of the DataBug will take place at a basic level via a console interface, similar to that used by the "telnet" program, and many serial devices. This will allow the DataBug to be controlled by any terminal program, such as those supplied with basic Windows installations.

At a higher level, the user will see a GUI or equivalent. They can select commands from a menu which will be sent as data packets to the DataBug. They will also contain an IP v.6 address, to link them to the correct DataBug port and meta-information about the packet itself, to allow high level error correction. If necessary, the packets can be encrypted.

It is intended that the packets would be human-readable, to allow simple debugging of installations. Encrypted packets would have readable meta-data, which would indicate the encryption mechanism used. A suitable format might be that used in NMEA-0183, a marine serial control standard:

```
$COMMAND,METADATA,DATA,CHECKSUM,<CR><LF>
```

The dollar symbol is an “escape character”, to differentiate the packet from standard console data. A “guard time” would be used as a further check, to ensure the packet had been sent automatically as a block, and not just typed by a human. The carriage return and line feed ensure the data stream can be viewed on a normal text terminal.

In order to have the correct commands available on the user interface, an XML file describing the necessary command sets is read in and used to generate the interface. XML is the eXtensible Markup Language. Using XML it is possible to define a Command Description Language for the DataBug (see Appendix). This can be used by an XML parser to convert the syntax of a command set into menus and options in a GUI.

In Table 2, the GUI would be generated as an HTML website by a server computer, which is also connected to a modem. In the case of a PC, this server could be the same machine as the client. Alternatively, the server could be accessed over a corporate intranet, possibly via “thin clients” such as mobile phones.

Packaging

The DataBug will be packaged in a polycarbonate case, injection moulded in two segments. The PC/104 stack will be screwed into posts in the base section. The case will be robust enough to survive light knocks without damage. It will also be sealed to give the necessary level of protection against ingress.

The case will be made from brightly coloured transparent plastic. This will differentiate the DataBug from other products, and help it stand out in catalogues. The colouring will also speed location of the DataBug in crowded equipment cupboard.

Security

A major concern in the DataBug software design will be security. The devices which it controls may be mission-critical their owners, who will wish to be assured that there is no possibility of unauthorised or accidental access. The server will have built-in authentication, to ensure that only legitimate users are allowed to access the web-based interface.

There will be a second layer of password checking (transparent to web interface users) at the DataBug itself. This prevents hackers accessing the equipment directly, and allows valid, power users to bypass the GUI interface.

Once a connection is established, data transfer could be encrypted, to prevent interception of confidential data. The DataBug could be set up to require an encrypted connection, to act as a further barrier to unauthorised use of the connected equipment.

Part of the DataBug's security will also derive from the fact that it is a point-to-point system, and the data passing through it does not go via the open Internet.

Manufacture

CAD/CAM methods will be used where possible. This reduces the lead time on production, and makes it easier to implement any changes found necessary after the prototyping stage. CAD/CAM would mainly apply to the design of the I/O board, and the moulds for the case. The latter would be produced from the 3D visualisation model of the design.

The Modem and CPU Board would be bought in as pre-assembled COTS components. The custom I/O board would be manufactured out-of-house by a PCB bureau service.

Construction of the DataBug is intended to be as quick and simple as possible. It would be performed in the following manner:

1. The PC104 boards are stacked together and screwed onto the base. There will be no flying leads to connect, or extra soldering to perform.
2. The top of the case is snap-fitted onto the base.

If used, the network module would be assembled similarly, but without screws. It is predicted that each assembly could be completed in under two minutes. Required cables could be built in pre-made, or speedily assembled using crimp-on, insulation displacement connectors.

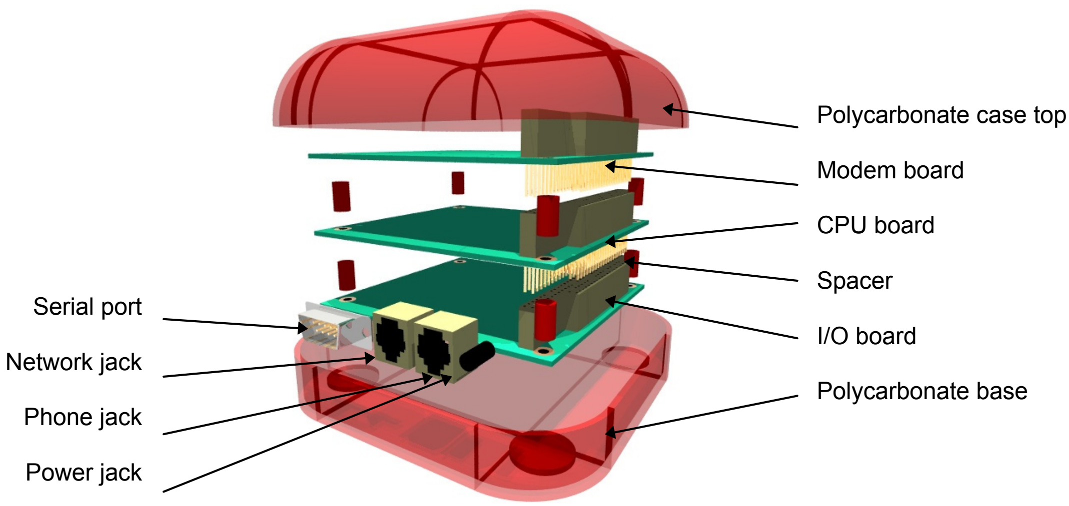


Figure 4. Exploded diagram of the DataBug.

Testing

Since the DataBug has not been realised, testing has been limited to proving various concepts associated with the design.

Physical Implementation

The physical dimensions of the components were used to construct a 3D virtual model, to allow visualisation of the design. This model allowed the designers to check that all parts would actually fit together, and verify that the working drawings were accurate. The modelling system used was POV-RAY.

Results from the modelling were incorporated in subsequent iterations of the design. In a production environment, the 3D model would also have been used to provide the CAD/CAM design of the mould for the DataBug's case.

Software

Software concepts were successfully tested by building a website to emulate the operation of a system incorporating the DataBug. The website demonstrated the following:

- Description of command sets by XML markup.
- Dynamic generation of a user interface incorporating several command sets.
- Interaction between an HTML interface and a Java program.
- Simulated interaction between a user and the DataBug, using either a console window or HTML interface.

The above is available for viewing via the Internet at the URL, http://www.dur.ac.uk/j.r.inge/bug/

Cost

Various costs have been identified for manufacture of the DataBug:

One-off costs:

Prototyping of PC/104 I/O card	€200.00
Prototype polycarbonate shell	€2500.00
Injection moulding tooling	€1000.00

Per unit costs:

Estimated on a production run of 1000 units.

AT/3i PC/104 CPU board (EuroTech S.p.A)	€350.00
Minimodem OEM #8410 PC/104 modem board (Aprotek)	€160.00
PC/104 I/O board	€20.00
Polycarbonate shell	€5.00
Fittings	€5.00
Assembly	€0.30
Packaging	€1.00

With a 20% profit, but neglecting company overheads, the selling price of the DataBug will be around €650, or £400.

Evaluation

The DataBug design fulfils all its design criteria. It could be manufactured at a cost of approximately £2000 for a prototype or £400 per unit in a production run of 1000 - markedly cheaper than a PC installation. The DataBug would also have lower impact on an installation in terms of both space and power consumed.

The user interface has been proven in concept and allows intuitive control of remote devices. It can be adapted to suit any type of serial equipment through XML command sets. These are simple enough that, if not available from the device manufacturer or DataBug company over the Internet, they could be written by anyone who has set up a website.

The DataBug could be easily improved or adapted in a number of ways. The standard PSTN modem could be swapped for an ISDN or xDSL modem with no major impact on the rest of the design. It could also be replaced by a GSM modem, to allow portable use. Such modems are already available in PC/104 format.

Adapters could be produced to allow use with other serial formats than RS-232, e.g. USB, RS-422, 423 or 485. A USART could be added to allow synchronous communications. These modifications are not technically difficult, but their applications are often even less standardised than for RS-232 links.

The DataBug software could also be adapted for use over the Internet by “e-Service” companies, or be embedded in other products to add functionality.

Conclusions

The DataBug system will allow remote control of serial devices in a way that is simpler and more cost effective than any existing system. It will add value to other, more expensive, equipment by reducing operating costs and increasing the range of locations in which they could be used.

There are potential applications for this technology in diverse areas such as telecommunications (routers, base stations), the utilities (pumping stations, switch houses) building management and the automotive industry (remote diagnostics). In fact the DataBug or a derivative system could be used in almost any remote control application.

The design is built on simplicity and achieves low cost through a double strategy. Cheap, proven hardware is combined with an intuitive user interface to give a system that is quick and simple to install, configure and use.

The simplicity of the DataBug and its modular design means that the system is easy to manufacture. Equally, design upgrades and changes can be implemented with little extra development cost.

In conclusion, the DataBug system is a workable concept which fulfils its intended ends, and could be easily adapted to a range of other situations and uses.

References

[1] "PC/104 Specification," PC/104 Consortium, March 1992, revised June 1996. Latest issue (Version 2.3) is http://www.pc104.org/specs/pc104-23.pdf

[2] ISO 9001:1994, "Quality systems -- Model for quality assurance in design, development, production, installation and servicing." Edition 2, ISO subcommittee TC 176/SC 2, International Standards Organisation, 15 November 1996.

[3] BS EN 60529:1992, "Specification for degrees of protection provided by enclosures (IP code)." British Standards Institute, 31 January 1992.

[4] "Council Directive 89/336/EEC of 3 May 1989 on the approximation of the laws of the Member States relating to ElectroMagnetic Compatibility", The Official Journal reference is OJ No L 139, 23. 5. 1989, p. 19. Directive amended by Directive-91/263/EEC (OJ No L 128, 23. 5. 1991, p. 1).

[5] "Council Directive 73/23/EEC of 19 February 1973 on the harmonization of the laws of Member States relating to electrical equipment designed for use within certain voltage limits." The Official Journal reference is OJ No L 77, 26. 3. 1973, p. 29.

[6] "Directive 99/5/EC of the European Parliament and of the Council Relating to Radio Equipment and Telecommunications Terminal Equipment and the Mutual recognition of their Conformity", 7th April 1999, Official Journal of the European Communities. The Official Journal reference is OJ L 91, Vol 41 of 7th April 1999.

[7] BS 6320:1992, "Specification for modems for connection to public switched telephone networks and speech band private circuits run by certain public telecommunications operators." British Standards Institute, 31 October 1992.

[8] BS 6328-1:1985, "Apparatus for connection to private circuits run by certain public telecommunication operators. Specification for apparatus for connection to speechband circuits", British Standards Institute, 31 October 1985.

[9] BS EN 41003:1999, "Particular safety requirements for equipment to be connected to telecommunications networks." British Standards Institute, 15 May 1999.

[10] "BABT Certification Regulations", Issue 4, British Approvals Board for Telecommunications, 1 August 1998.

[11] BS 6312-2.2:1997, "Connectors for analogue telecommunication interfaces. Sockets for use with plugs specified in BS 6312:Part 1. Particular requirements for fixed socket-outlets used in permanent wiring installations." British Standards Institute, 15 April 1997.

[12] V.21, "Recommendation V.21 (11/88) - 300 bits per second duplex modem standardized for use in the general switched telephone network," International Telecommunications Union, November 1988.

[13] V.22, "Recommendation V.22 (11/88) - 1200 bits per second duplex modem standardized for use in the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits," International Telecommunications Union, November 1988.

[14] V.22bis, "Recommendation V.22 bis (11/88) - 2400 bits per second duplex modem using the frequency division technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits," International Telecommunications Union, November 1988.

[15] V.32, "Recommendation V.32 (03/93) - A family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits," International Telecommunications Union, March 1993.

[16] V.32bis, "Recommendation V.32 bis (02/91) - A duplex modem operating at data signalling rates of up to 14 400 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuit," International Telecommunications Union, February 1991.

[17] V.34, "Recommendation V.34 (02/98) - A modem operating at data signalling rates of up to 33 600 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits," International Telecommunications Union, February 1998.

[18] V.42, "Recommendation V.42 (10/96) - Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion," International Telecommunications Union, October 1996.

[19] V.42bis, "Recommendation V.42 bis (01/90) - Data compression procedures for data circuit terminating equipment (DCE) using error correction procedures," International Telecommunications Union, January 1990.

[20] ISO 9000-3:1997, "Quality management and quality assurance standards -- Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software." ISO subcommittee TC 176/SC 2, International Standards Organisation, 18 December 1997.

[21] CSS, "CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., 12 May 1998. The CSS2 Recommendation is: http://www.w3.org/TR/1998/REC-CSS2-19980512. The latest version of CSS2 is available at: http://www.w3.org/TR/REC-CSS2.

[22] XML, “Extensible Markup Language (XML) 1.0”, eds. T. Bray, J. Paoli, and C. M. Sperberg-McQueen. 10 February 1998. Available at http://www.w3.org/TR/REC-xml.

[23] HTML, “HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 8 July 1997. Available at http://www.w3.org/TR/REC-html40/. The Recommendation defines three document type definitions: Strict, Transitional, and Frameset, all reachable from the Recommendation.

[24] “The Art of Electronics”, P. Horowitz, W. Hill, Cambridge University Press, 2nd Edition, 1989, pp 723-726.