The Systems Journal

ICL

Special Theme — Electronic Business

Volume 14 Issue 2 *Spring 2000 Edition*

ICL Systems Journal

All correspondence and papers to be considered for publication should be addressed to the Editor.

The views expressed in the papers are those of the authors and do not necessarily represent ICL policy.

Published twice a year by Group Technical Directorate, ICL, Stevenage.

2000 subscription rates (including postage & packing):

	UK and Europe	Rest of World
Annual subscription	£72	$120
Single issues	£43	$72

ISSN 1364-310X

ICL Systems Journal

Volume 14 Issue 2

Contents

The Electronic Commerce Revolution

Dick Emery

ICL, Kidsgrove, UK

Abstract

Electronic commerce is no nine-day wonder. It has now earned itself a deserved place in the computing hall of fame alongside the spreadsheet, the relational database and the COBOL compiler. Electronic commerce routinely mechanises the supply chains of big retailers and manufacturers. It is sending shock waves through retail banking and has given rise to unprecedented growth in technology stock markets. Electronic commerce is becoming the method of choice when purchasing books and records. It is here to stay. As electronic commerce relies fundamentally on advanced information technology, it is appropriate to consider the subject in some depth in a technical journal. At the same time, it is important to reflect on the social and business implications. After all, it is through its non-technical aspects that most people will feel the benefits of electronic commerce. These aspects also play a crucial role in deciding what will ultimately prove a success or a failure in the market. We have to use a wide perspective if we are to understand correctly the electronic commerce revolution.

Defining Electronic Commerce

There have been many attempts to define precisely what we mean by "electronic commerce" — doing business on-line, trading across networks and so on. But the urge to achieve brevity usually results in an opaque outcome. It is like describing in one sentence "transport", as we now understand it, to a citizen of mediaeval England and expecting to impart some enlightenment. It is probable that electronic commerce has too many dimensions and too much capacity to develop in surprising ways to make any one-sentence definition even remotely useful.

But that must not deter us from marking out some of the boundaries. We can certainly make some assertions about electronic commerce which sharpen our understanding of what it is. The ones that appear below apply to the categories of electronic commerce which we consider in the following articles in this issue. In truth, the result excludes some activities which informed commentators often consider to be legitimate forms of electronic commerce. Our narrower choice here is deliberate. The intention is to condition the reader's mind for the better understanding of what follows rather than to enter into learned debate about whether our choices are actually right or wrong.

- Electronic commerce relates to the exchange of goods or services for value. This leads to the exclusion of all activities which do not contribute to the creation and execution of contracts. This suggests that electronic commerce does not take place *within* an organisation since contracts operate *between* organisations and individuals. For completeness, we take the view that a citizen submitting information, and thereby receiving entitlement to a state benefit, qualifies for inclusion.

- Electronic commerce relates to remote parties communicating across open, digital networks. This causes the exclusion of systems on private networks between consenting parties. This has the effect of sharpening the challenges of achieving integrity, security, privacy, authentication and auditability.

- Electronic commerce takes place between two parties where one (or both) must be a device not in the synchronous control of a human. This excludes voice telephony and facsimile since both are essentially passive requiring human participation in order to lead to contracts.

When we apply these assertions, we find ourselves firmly in the thrall of Internet-based systems.

Payment, where relevant, is typically by credit card. There may be electronic fulfilment for software purchases or entertainment downloads. But equally there may not. In other words, these assertions appear correctly to locate electronic commerce precisely where events are moving most rapidly and where current attention is most firmly fixed. This vindicates the validity of the assertions for our purposes.

Size and Growth

All observers agree that electronic commerce (after applying the assertions from above[1]) is in its early days but is growing very rapidly. The debate is whether it is going to be big, very big or enormous. For reasons of disparity in definitions, timing, geographies and sectors, observers and forecasters find it difficult to arrive at any useful agreement as to how big electronic commerce already is or how big it will become, at least within a factor of ten. For example, they have reported annual growth rates ranging from two to four plus. But perhaps this is typical of something this young, growing this quickly.

A cold, hard look, taken from the perspective of the north of Europe, suggests that electronic commerce is just beginning to become part of the lives of its early adopters. Excellent examples are the emergence of Internet banking, share trading and book buying. There are now people who would be inconvenienced if the Internet and electronic commerce were to stop functioning — a sure indicator of adoption. But at present the inconvenience would strike very few and recovery would be rapid. On this test, electronic commerce only accounts for a fraction of a per cent of economic activity. It is far from being "the way we do things" as, for example, car driving is now.

Nevertheless, it is easy to see how this situation is changing. When so many traditional advertisements display a URL and newspapers have weekly supplements on the Internet, including electronic commerce, there has to be something in the air. We are surely able to envisage, at least approximately, how far things could develop. It takes little imagination to see how Internet banking, Internet insurance selling, Internet travel purchasing, Internet tax returns and the like could become norms for many citizens. Examples of such systems are already capturing a share of the market at a considerable pace. If the top third of UK households were to spend just £2000 per annum across the Internet for these purposes then the total transacted value would reach £16 billion, perhaps 3% of consumer spending. This could easily happen within relatively few years, perhaps as few as three. The United States market is arguably crossing the 3% threshold at the time of writing.

We can deploy arguments in favour of a growth of this amount by a factor of five over the following years, reaching say £80 billion for the UK, but no one knows how long this will take or what new services will spring into being to attract the consumer's pound, euro or dollar. However, behind the scenes electronic commerce will support the enterprises which manufacture, supply and serve the active consumers. Some argue that this could be as much as ten times the transacted value which the consumer sees. We might cautiously envisage an economy which becomes as reliant on electronic commerce as it is today on road transport. Many new ways of doing business, currently unimaginable, will arise just as they did when the internal combustion engine replaced the horse. Human characteristics being what they are, we can expect the aftershocks of the electronic commerce quake to be with us for twenty or thirty years into the future. We still have a long way to go.

Major Segments

The Internet makes possible point to point communication between individuals, organisations and devices of all kinds. Either or both ends of the link can be static or mobile. Through the power of optical fibres, satellites, cellular radio and more, we can realise the "any time, any place, anywhere" proposition. Where the "wired world" goes, electronic commerce goes too.

Almost all commentators agree on a useful segmentation of this apparently limitless space.

[1] Without this limitation, one could claim that electronic commerce already transacts $1 trillion plus every day in the form of foreign exchange, capital markets, credit card clearing and so on. Similarly Electronic Data Interchange (EDI) already accounts for daily, large transfers of value. All these modes of electronic transaction are far from new and are not the cause of the present popular excitement.

They recognise:

- Business to consumer (B2C) applications. All Internet retail (cybershops, cybermalls, merchant sites, retail banking, ...) falls into this category and so does most entertainment.

- Business to business (B2B) applications. This covers corporate purchasing, supply chain management, distribution management and much more.

- Government to citizen applications. This is sometimes known as electronic government and relates to government interactions with both citizens and business. Some title it e-government.

- Individual to individual applications. Although poorly developed at present there is a vast potential for individuals to trade with other individuals using the Internet as the medium.

This segmentation omits what could well become the most populous segment of all — device to device. Those who look further into the future see a world in which almost every device — car, refrigerator, heating controller, video recorder, burglar alarm, personal digital assistant (PDA) — will be connected to the Internet, probably by radio. These connected devices will work silently to make human life more pleasant. For example, one's car could report the diagnostics of a fault condition to the local service centre, order the necessary replacement parts, negotiate a suitable service date and time with the service centre and one's PDA, and only seek final human confirmation for the complete arrangement once this had been negotiated. In analogous ways, the refrigerator could replenish itself, the intruder detection system could refer suspicious conditions to a security centre and so on.

Since there are numerically many more devices than humans, at least in developed countries, the potential for silent transactions between devices is much greater, in volume terms if not in total transacted value, than between humans and organisations as we currently know them. Through the exploitation of Wireless Application Protocol (WAP) and the Bluetooth development of standards for wireless device connection, we are now at the leading edge of an exciting and intriguing explosion. No one knows where it will go.

Holistic Approach

The constituents of a successful electronic commerce solution come from many disciplines. Information technology is certainly necessary but far from sufficient. Even the best technical execution of a solution cannot compensate for an attempt to sell something which no one wants at a price which few can afford. Equally, any solution which breaches the law will not operate for long. All the contributing factors have to be under control if failure is not to be a real risk. However, there is no combination which guarantees success.

We can consider the factors in four major categories:

- **Business proposition**
  Every solution exists for a purpose. This must be effective from strategic, business and marketing perspectives. There has to be a sound business model which has the potential to deliver value and profit. No amount of competence in other fields can compensate for a flawed vision, be it of potential customers, products, services, competition or prices. These are the issues which confront every entrepreneur, investor or business manager whether the commerce is electronic or not. Being electronic brings some new issues into consideration, but it removes none of the old ones.

- **Regulatory compliance**
  The solution must comply with all the necessary regulation, legislation and, in most cases, best practice guidelines. In the present climate of uncertainty and rapid change, this presents a significant challenge to designers. Nevertheless it is an unavoidable fact of life that failure results in early termination of the solution. But there are also more subtle features relating to the solution's longevity against a shifting legislative background, to the potential for litigation in case of disputes and to the avoidance of fraud and other misuse.

- **Operational attractiveness**
  The solution must be attractive, distinctive, pleasant and easy to use for its target population and, where relevant, engender positive loyalty. Much of this arises out of

designing both the functions and the style with the typical, target user in mind. For IT experts this can prove to be a challenge too far. The disciplines of ergonomics, industrial design and graphic design all have their roles to play here.

- **Implementation effectiveness**
  The solution must work correctly to its design. This is the traditional arena of IT experts, project managers and operations managers, supported by graphic artists and copywriters. While, say, the Internet brings new knowledge domains which play their roles in creating success, there is a great deal here which relies on the established best practices of IT more generally.

Social Implications

It is far too early to forecast how and where electronic commerce will alter the lives of the world's citizens. Only with hindsight can we now speak with any confidence about the social implications of telephony, aviation and television. The effects, when taken over decades, can be remarkable. Who would have thought that aviation would lead to the building of large concrete buildings (hotels) along the coast of southern Spain? Or that the equity value of entertainment companies would exceed that of steel, coal and shipbuilding combined? What we can forecast with some confidence is that electronic commerce will have a dramatic effect but we can only vaguely discern what it will be.

One early effect is likely to be a reduction in prices as electronic commerce lowers costs and opens up competition between previously isolated markets. We can already observe the effects in retail banking and in easily transportable goods such as CDs. This is clearly only the beginning of what could so easily happen. There are some who argue that the effect on retailing will be at least as great as the transformation of the traditional High Street grocer into the out-of-town hypermarket. Electronic commerce could well change radically our shopping habits.

As electronic commerce is no respecter of international boundaries, its practitioners now confront the issues of cross-border, consumer trade with all the concomitant problems of taxation, consumer protection, contract formation, data protection and court jurisdiction. Electronic commerce is amplifying national differences of approach which have always existed but formerly rarely mattered. Now they do matter and there are few easy solutions. For example, no national treasury wants to cede its consumption tax revenues to other nations which happen to enjoy more success as Internet retailers. We can confidently expect these issues to rumble on for many years to come while national leaders slowly accept the inevitability of convergence on a world scale.

One class of individuals is likely to gain disproportionately from electronic commerce — those with low mobility through disability. Electronic commerce has the potential to bring the world's shops and the world of work and entertainment into the home. Just as the car removed the challenge of buying in bulk, or at least weekly, for those with below average physical strength who had to walk or take public transport, so electronic commerce can remove the need for any mobility for some major types of activity. This also leaves aside the non-commercial uses of the Internet for learning and socialising.

Looking further ahead, some are predicting that organisations who provide the physical fulfilment services will dominate and reshape themselves to provide co-ordinated delivery of whatever is then available from all sources, rather than from just one, rather like letter post but with preferences over timing to match work and life patterns. Some predict the dominance of electronic versions of money — say, a pan-global currency for use in cyberspace — to the extent that central banks will lose control of their money supplies. Some predict the adoption of a universal smart card for use in authenticating personal identity when undertaking electronic commerce. However, such dramatic forecasts are purely speculative and could just as easily be false. The one certain factor present in all such forecasts is that electronic commerce is going to make a big difference to us all.

Conclusions

All informed observers unanimously express their confidence in electronic commerce's ability to revolutionise major aspects of the world in

which we live. While accepting this postulate, we have to remember how much collateral change is happening and has to happen in related fields. Electronic commerce may challenge IT, but it also challenges legislation, taxation, traditional commerce, national identities, social expectations and so on and so forth. In all revolutions, the long-term effects rarely match the aspirations of the inspiring revolutionaries. The articles which follow look at the present and the near term. They excite our interest with what is new today. But we should prepare ourselves to be excited for a good while longer as electronic commerce will surprise us over and over again.

Biography

Dick joined ICL immediately after graduating from Cambridge University. He spent the first half of his career in technical roles associated with sales and major customers. He became a national authority on operating systems and contributed managerially and technically to the early life of the ICL 2900 Series mainframes. Dick then moved through sales management into marketing. He rose first to the position of Marketing Manager for ICL Mainframes world-wide. He then took marketing responsibility for ICL's systems integration capability and in particular the OPENframework methodology. He also worked within related areas such as Open Systems and Intranets. He is currently responsible for creating and maintaining ICL's corporate strategy for electronic commerce. He represents ICL's electronic commerce viewpoint on a variety of trade and government activities as well as advising customers on their electronic commerce strategies.

Content Management Solution Kit (*Athens*)

Paul Duxbury

ICL, Kidsgrove, UK

Abstract

ICL's new Content Management Solution Kit (*Athens*) is intended to expedite dramatically the development of e-Business solutions, by providing for most of their content-related needs. Like all ICL solution kits it is designed to be used by ICL project teams, but many of its features and benefits will also be visible to customer roles such as authorship and administration.

Athens offers a complete solution for stand-alone publishing sites, comprising functions like content storage, administration, interchange, and publishing. Alternatively, *Athens* may be combined with other solution kits and external components through its many integration facilities: at its heart is the industry standard WebDAV protocol with corresponding API, and virtually all the components of *Athens* can be substituted.

A key feature of *Athens* is its ability to handle content stored on many different types of physical medium, including various relational databases, filesystems, volatile (RAM) memory, and composite XML documents. Multiple media types may be accessed concurrently and uniformly, and custom sponsors may be developed to integrate external or "legacy" data. Publishing features include powerful Web-page construction and personalization tools, with multi-device support. An XML-based document-interchange mechanism is also provided. All the facilities of *Athens* are offered through user-friendly, Web-based administration interfaces suitable for both locally and remotely hosted services. High performance & scalability are major objectives.

Introduction

This paper outlines the main concepts and features of ICL's Content Management Solution Kit (CMSK), known also as *Athens*, and should provide useful reading for anyone about to embark on any e-Business project, since all such projects include content in some form.

In summary, Athens has the following principal features:

- It provides most of the content-related needs of e-Business solutions (storage, management, page-construction, publishing etc.)
- It provides a complete solution for stand-alone publishing sites ...
- ... or use with other solution kits and components through its many integration facilities
- It provides flexible Content Storage Features — database, filesystem, legacy etc.
- It has user-friendly Administration Interfaces
- It includes powerful Page Construction, personalisation and Publishing Tools
- It offers high Performance & Scalability.

These features are described in some detail in the body of the paper. However, some more basic questions are considered first, such as, "What is content" and "Who needs Content Management?"

What is "Content"?

"Content" means any information or goods which are delivered electronically to an e-Business consumer, either directly or indirectly. Originally, content was embodied in HTML pages and their associated images, and delivered directly to users

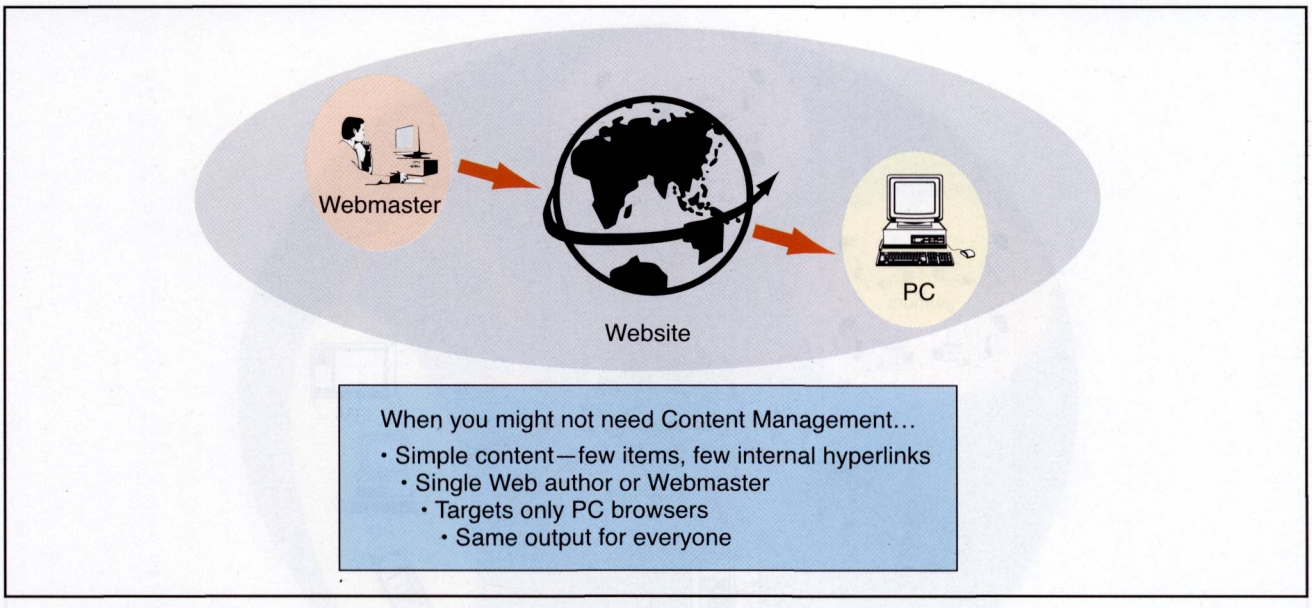


Figure 1: Content Management NOT required

through the World Wide Web. However, the advent of more general e-Business applications has led to a corresponding generalization in the definition of content to include, for example:

- Web pages & images
- Multimedia files (for example, audio & video clips)
- Streaming media
- Shopping catalogues
- "Soft" or "digital" goods for sale (e.g. downloadable music and software)
- Even "functionality", as embodied in CGI scripts and their modern equivalents.

Furthermore, much content is no longer stored in the exact form in which it will be delivered to a consumer, but rather as "raw" data used both to construct pages and to drive application processes. For example, the price of a product is needed both in the catalogue pages shown to users, and in the business logic used to handle customer orders, so it makes sense to store it separately.

What is "Content Management"?

"Content Management" may broadly be defined as a collection of facilities for:

- the introduction, storage, administration, exchange and delivery of e-Business content in all its forms...
- ...with appropriate accuracy, efficiency, usability, automation, flexibility and personalization.

Who needs Content Management?

For simple e-Business applications, like small Websites, Content Management may not be much of an issue. There are plenty of excellent tools to allow a single "Webmaster" to author and/or install HTML pages and images in a conventional Webserver. Assuming that there is not too much content, that it does not change too frequently and that it is appropriate for everyone to receive exactly the same version of each page irrespective of their preferences and browser device, then this approach may be perfectly adequate (Figure 1).

However, as soon as any of these operational parameters are relaxed, complexity can grow rapidly. For example, suppose each Web page is required to include a list of links to others on the same subject, or by the same author. Each time a new page is introduced or an old one removed all the other pages will need to be revised. This is easy if there are only a couple of dozen pages. But what if there are 10,000? And what if several hundred pages change every day? So much material cannot normally be authored and controlled by one "Webmaster" and a substantial

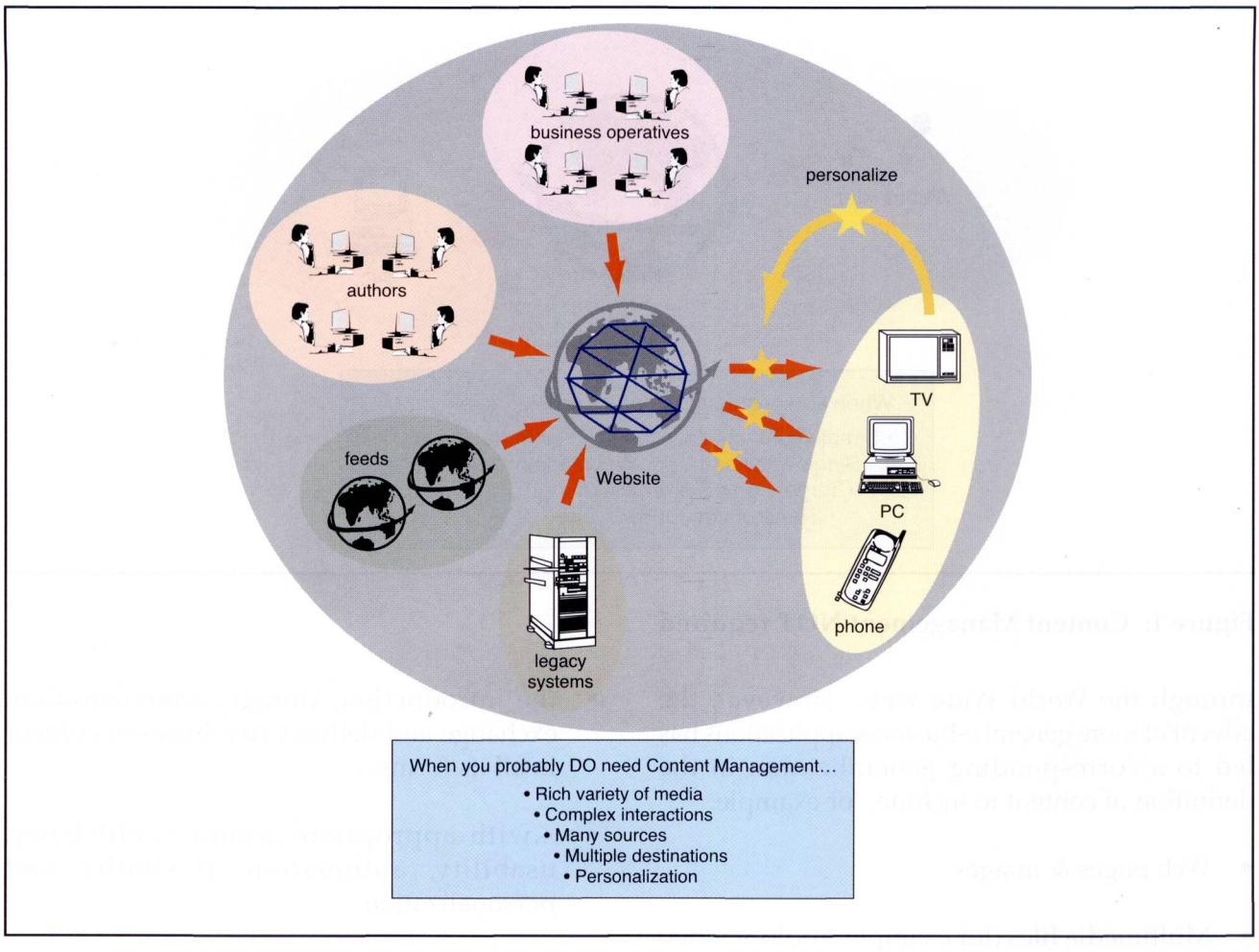


Figure 2: Content Management required

team may be required for the purpose. How is it ensured that the members of such a team work effectively together and maintain the consistency of both individual pages ("house style") and entire collaborative sites?

Then, in many cases, e-Business applications require integration into more conventional business processes. This means that site contributors and administrators may not be "Web" people, or even IT people, but may be journalists, editors, marketing staff, telesales and customer service operatives. Each will expect to have content management facilities tailored to his or her own needs — for example, a simple forms-based interface for entering a classified advertisement for a used car.

Another major factor is access to material originating from outside the specific application, whether it be legacy data from an existing orthodox business, or dynamic feeds from some news agency.

There are many other similar considerations concerning performance, scalability, security and so on. A specific mention must be made, however, of targeting, which is becoming ever more a requirement and expectation — it is one of the unique capabilities of e-Business applications which distinguishes it from orthodox business operations.

By targeting we mean delivering different content to different individuals according to their preferences and/or means of connection. For example, e-Business applications can now be accessed by a wide range of devices — not only various PC-based browsers, but also mobile phones, pagers and interactive TV (Figure 2). The protocols used may not be compatible, and it is not always appropriate to send precisely the same content to, say, a mobile phone as it is to a 17 inch monitor. Thus multi-device publishing is frequently required. In addition, personalization of content based on a user's preferences (declared or deduced) is also becoming a major tool in attracting and retaining visitors.

How does Content Management help?

As defined in this document, Content Management helps at all stages of the design, construction and operation of an e-Business application, such as a complex Website. To understand how, the functional model illustrated in Figure 3 will be used.

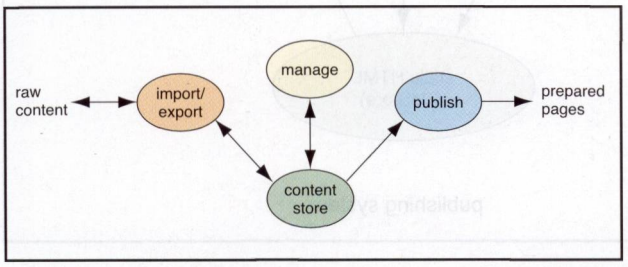


Figure 3: Content Management — Functional Model

At the centre of a Content Management System is the Content Store. According to the application and the nature of the physical storage, this may hold both content and metadata (see later) and the content itself may consist of raw reusable data or prepared Web pages.

The remaining subsystems are all clients operating on this storage service. The first involves installing content in the content store, and retrieving it for modification or export to other stores. A variety of mechanisms is required to suit all likely content sources and destinations. Next comes the user interface for managing the content already in the Content Store. Functions include moving, replicating and tagging content with properties, as well as controlling the other subsystems. Finally, content needs to be published for consumers. If the content consists of ready-made HTML pages, publishing may consist simply of delivery via a Webserver. However, for raw content some form of page construction process is needed. *Athens* will be seen to be well equipped in each of these main functional areas.

The following two examples illustrate the use of the model for different styles of application (in this case Websites).

1. A conventional Website, where the content store is an ordinary filestore containing prepared HTML pages, import/export may use FTP, publishing may use a standard Webserver, and management may be effected through a standard Webserver administration and/or a file-manager.

2. A dynamic Website, where the content store is a relational database holding raw data, import/export and management may use a forms-based data-entry interface, and publishing may use some dynamic page construction mechanism such as CGI or ASP, or the template system provided with *Athens*.

Though these examples are representative of many sites, the model presented here is admittedly very simple, and real-life production processes are potentially much more complex. For example, they may involve many stages on different physical hardware platforms. However, these can often be represented by combinations of the simple model as shown in Figure 4, where a dynamic Website prepares pages for storage on one or more conventional Websites.

Some Content Management Techniques

So how do Content Management systems typically solve the problems posed by large and/or complex e-Business content systems? This section outlines a few of the main underlying principles.

Template-based page construction

As more and more Websites start to contain functionality as well as static information, in some cases fronting major enterprise applications, so it is becoming common to generate Web pages independently for each user, dynamically on request. This is necessary so that results of queries or calculations can be inserted into the pages, and so that personalization can be effected.

Traditionally this meant that building Web pages became a development programming task. "CGI" programs were written to perform application functions and output HTML responses. However this meant that programmers were also responsible for the look and feel of the site, normally the province of graphic designers. Also, the simplest change to site design had to go back to the development programmers who created it.

To help this situation, some form of "template-based rendering system" is now often included

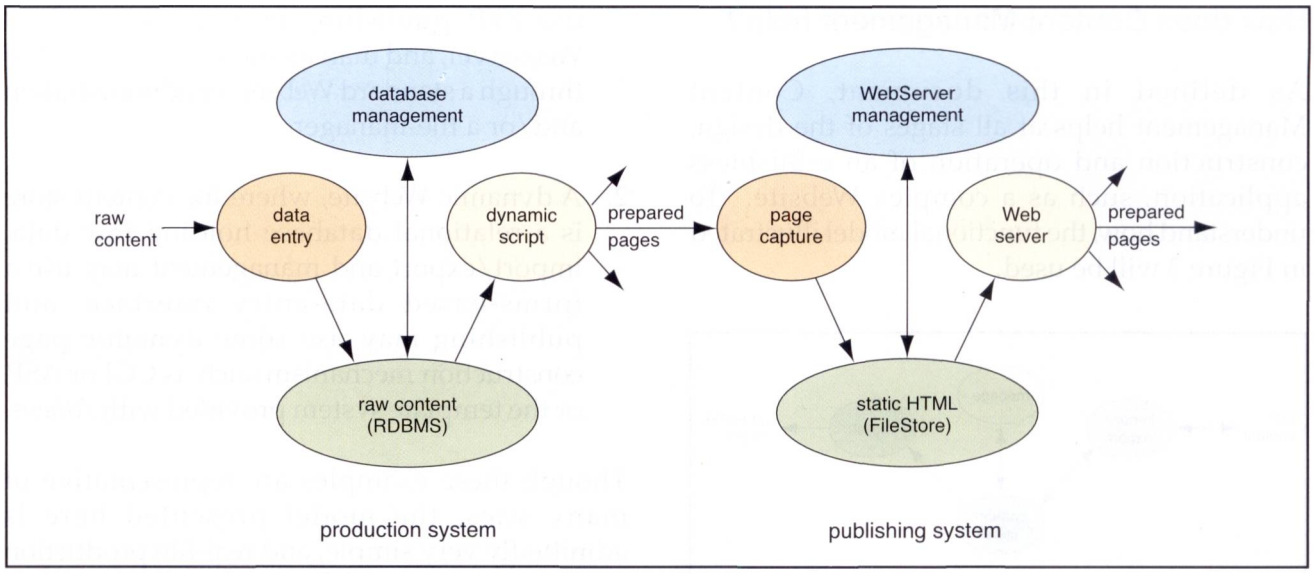


Figure 4: Content Management — Dynamic Web Sites

in content production systems. Here, a graphic designer generates HTML for the look and feel of a site, but leaves "holes" into which dynamic information can be placed by programmers. This is known as a template. The process is illustrated in Figure 5.

Although almost mandatory for dynamic page generation, template-based rendering can in fact also greatly benefit the production of "static" pages, especially where many pages are expected to follow the same basic layout. In such cases

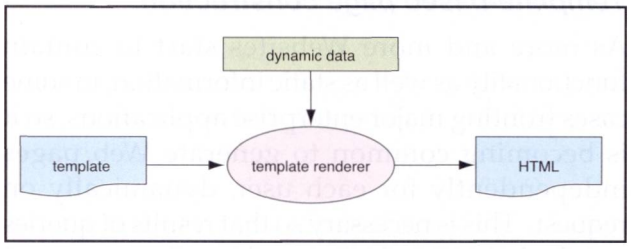


Figure 5: Template-based Rendering

the templates capture these layouts, and help impose house styles. Furthermore, changes in look and feel need only be applied to the templates, not to all the pages that are rendered by them, and alternative templates can be created for use with different client devices. Some systems also provide mechanisms for generating hyperlinks automatically at the page rendering stage, to ensure their consistency (see later).

The basic advantages of template-based rendering are that it:

- Separates programming and graphic design aspects, and hence the skills needed
- Allows standard layouts, navigation, and house-styles to be imposed easily
- Makes changes to look and feel easier, since only the templates need altering
- Allows alternative renderings for different devices, by supplying multiple templates for the same content
- Allows automatic construction of hyperlinks in some cases.

Metadata

Metadata is "data about data". Its use is at the heart of many content management procedures.

Many data storage systems and applications use metadata, although they may not label it as such. For example: files in a traditional filesystem are accompanied by file properties (e.g. size, access rights, modification date); word processing documents may be accompanied by "property sheets", recording authorship and subject matter; a search engine may generate indexes for documents to speed up searching. All these are examples of metadata. However, their storage, accessibility, and usage tend to be restricted and non-uniform.

The ability to store and manage arbitrary sets of metadata against any item of content in a

uniform way is a great advantage for both e-Business and conventional applications, so content management systems which support it fully have much enhanced potential. Some examples of metadata which might be attached to an object, and example uses, are:

- Subject keywords, used to deduce a visitor's interests and provide him or her with more of the same (one form of personalization)
- Authorship information, used to record ownership and route feedback comments
- Document status (e.g. in-preparation, complete, approved, withdrawn), used to control workflow processes
- Rendering parameters, e.g. what template to use, background images etc.
- Relationships with other items, used to construct hyperlinks.

Metadata may be stored embedded in text files (e.g. as HTML metatags), or in a database, or in application-specific sections of binary files. Adding metadata is sometimes known as "tagging".

Database Storage

Conventional Webservers store their content in a filesystem, as prepared HTML files and images. However, the ability to use relational database systems instead has a number of advantages:

- Easy to hold related metadata (e.g. as additional columns in tables)
- Can exploit physical scaling, resilience and management features associated with RDBMSs
- Relatively fast keyword search
- Readily accepts structured raw data needed for dynamic page construction
- Can model object relationships, used (e.g.) to generate navigation and other hyperlinks.

Ideally a content management system will support RDBMS storage of both text and binary data, with arbitrary amounts of metadata, and without imposing a heavy database design and administration burden on site developers and operators.

Soft Hyperlinks

The defining feature of the World Wide Web is the *hyperlink*, which (with help from the HTTP protocol) allows any Web page to route visitors transparently to anywhere else on the globe. However, hyperlinks also present one of the biggest headaches when constructing and maintaining Website content. Generally, the more well-placed hyperlinks a site has — to related documents, to other site regions and features, to referenced external sites — the more usable the site will be, but, as the number of pages increases, the effort of maintaining these links (adding, removing or adjusting links as target pages are added, removed, or moved) can grow rapidly. In the limiting case, where every page is linked to every other page, growth would be exponential.

An approach taken by some content management systems is to generate hyperlinks automatically at page construction time, rather than hard-wiring them into Web pages. For example, if a template rendering system is offered, then information describing the desired target is inserted into the template, rather than the target URL itself. This will be called a "soft hyperlink".

A simple example is provision of a "document index" facility within a template. This instructs the renderer to insert a list of hyperlinks to all content items in a particular folder. Then, as content is added to or removed from the target folder, hyperlinks will be added to or removed from the current page automatically.

More refinement can be achieved using metadata. For example, hyperlinks may be generated to only those content items which have a particular subject field — maybe the same subject field as the present document, in which case the list will contain *related* documents.

Management of external links can be achieved by holding the target URL as a separate item of content, rather than encoding it directly into an HTML page. Within templates this item is used to build hyperlinks (<A> tags) field by field — the link text or image reference may also usefully be held as discrete items of content along with the URL. This construction may be repeated in

any other templates which require the same hyperlink. Then, when the link needs to be changed, only one small item of content has to be altered, not many complex templates.

These approaches can be combined, e.g. by adding subject keywords (metadata) to content items holding URLs, so indexes of related or personalized links may include external references.

Athens — Concepts and Features

The *Athens* Content Management Solution Kit itself will now be discussed. This is a fully featured Content Management and Publishing system designed to address the issues raised in the previous section and, thereby, to satisfy the needs of a wide range of e-Business applications.

The overall purpose and structure will be described in terms of the model presented earlier. Then each of the main subsystems will be discussed in detail. Finally, the integration features will be examined.

Overview

First the specific purpose, scope and architecture of *Athens* will be outlined.

Purpose & Scope

Athens is a fully-featured content management system designed to provide most of the general content-related facilities needed by ICL's e-Business solutions.

As a "solution kit", the "80-20" rule is expected to operate; i.e., unlike a conventional product visible to customers, *Athens* is targeted at ICL solution builders and is expected to save about 80% of the content-related costs which would be associated with a purely bespoke implementation. The remaining 20% (graphic design, legacy integration etc.) is necessarily bespoke.

The scope of the solution kit is rather wider than might be inferred from the title "Content Management", since it also includes many features relating to content publishing, information exchange, and application design. Since these features pervade all current e-Business solutions, the architecture provided also offers a more general basis for such solutions; i.e., it can serve as a high-value framework into which application-specific functionality (business logic) can be plugged.

A number of physical manifestations of this architecture are provided "out of the box". However, it is also intended to be equally well suited for hosting, as a logical layer, on top of other enterprise-strength application server platforms or middleware (e.g. COM, CORBA, EJB).

In summary, *Athens* is intended:

- To provide most of the content-related functions associated with e-Business solutions
- To be a Solution Kit
- To provide a framework for other application areas & solution kits
- To integrate well with other solution kits.

Highlights

For many regular publishing sites, as might previously have been accommodated by ICL's WebUpdate or Monterey, *Athens* is expected to provide a complete "out-of-the-box" solution. Facilities are provided for all the main functional areas associated with content storage, management, and publishing. There is usually, of course, an element of bespoke work needed to design a site's unique look & feel.

The individually supplied components have been designed to be as versatile as possible.

Central to the system is Content Storage. At the logical level, this provides a rich set of features for modelling most content needs, whether file-oriented, database-oriented, or "object plus metadata". At the physical level, several storage media are provided to match all likely requirements for cost, scalability, and flexibility.

Likewise, the management interface is intended to be user-friendly by offering a familiar "explorer" type of image, through a browser. Also, the template-based publishing system is intended to span both simple text-insertion capabilities suitable for hands-on use by customers, to sophisticated page construction and automation of administration tasks.

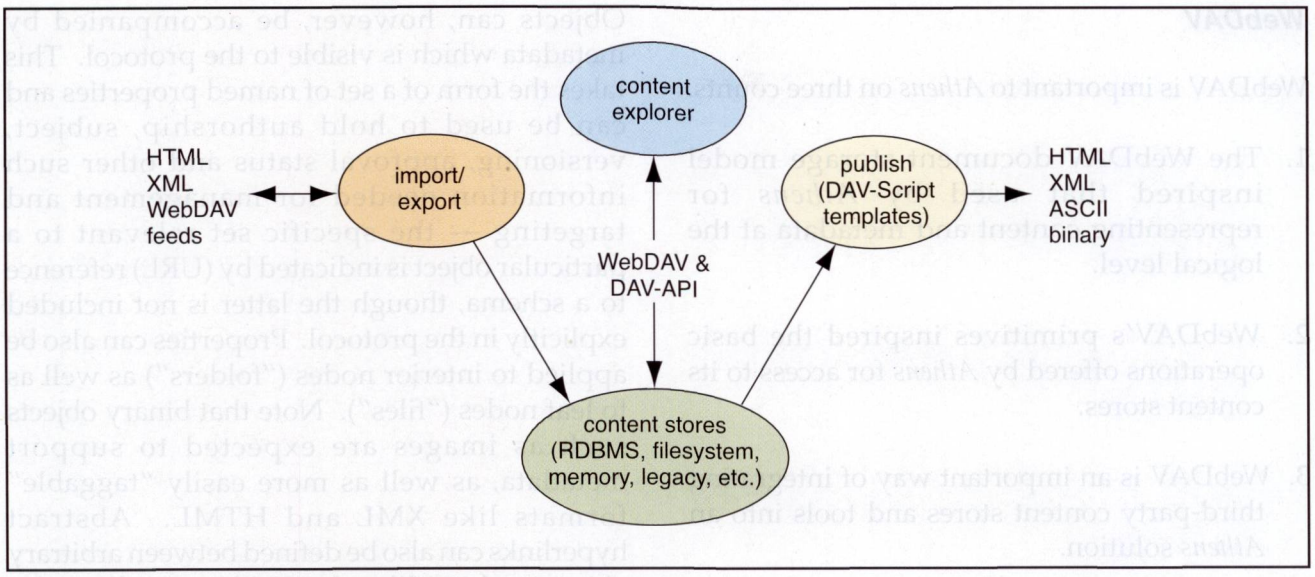


Figure 6: Athens Functional Architecture

But, despite such coverage, it is recognised that many solutions will have special requirements and/or need integration with other components. *Athens* is well supplied with APIs and protocols, including the industry-standard WebDAV, which enables *Athens* clients & servers to be supplemented or substituted as required. Custom content stores are a common technique for integrating legacy or external data.

Throughout, performance and scaling of solutions have been key considerations.

In summary, *Athens* offers:

- A complete Content Management and Publishing Solution
- Flexible Content Storage Features
- User-friendly Administration Interfaces
- Powerful Page Construction & Publishing Tools
- Wide Applicability through Rich Integration features
- High Performance and Scalability.

Athens Architecture

Using the simple model introduced earlier, *Athens* has the basic structure shown in Figure 6.

Taking the subsystems in turn:

1. For content storage we provide various physical media, including filesystems, databases, memory, XML and the ability to incorporate legacy or external data. These may be used in any combination
2. For import/export we have various facilities, both batch (e.g., XML-based bulk transfer), and interactive (e.g., browser file upload). Automation is provided by a built-in scripting language called DAV-Script (more commonly associated with template-based publishing, as below). Access to both local content storage and remote content sources is provided
3. For a management user interface we provide a browser-based "explorer" style of screen, with folding-tree views, cut-and-paste and similarly familiar paradigms
4. For publishing we provide powerful template-based page construction facilities, here using DAV-Script as the template language.

Finally, these subsystems need to communicate. The interface is modelled on the open industry-standard WebDAV protocol. This allows clients to modify content servers over the Internet or an Extranet and, therefore, lets third-party components be incorporated into an *Athens*-based solution. For efficient interaction within a single server, a Java API ("DAV-API") version is also provided.

WebDAV

WebDAV is important to *Athens* on three counts.

1. The WebDAV document storage model inspired that used by *Athens* for representing content and metadata at the logical level.

2. WebDAV's primitives inspired the basic operations offered by *Athens* for access to its content stores.

3. WebDAV is an important way of integrating third-party content stores and tools into an *Athens* solution.

WebDAV stands for "Web-based Distributed Authoring & Versioning". It is intended to allow open interworking between Web authoring/management tools and Web content stores, and is pitched at a high enough level to allow many possible implementations of each.

A typical proprietary protocol intended to be replaced by WebDAV would be the "Front Page Server Extensions" by which the MS Front Page and Visual Interdev authoring and development tools can update a Web server. Microsoft have endorsed such a replacement by including support for WebDAV in IE5 which allows the contents of Web folders to be manipulated. Client support is also expected to be included in Front Page, Visual Interdev and MS Office. IIS5 and Windows 2000 intend to offer WebDAV server support mapped directly on to Windows 2000 filesystem features.

WebDAV defines an abstract model of a content store, leaving it to server implementations to map this model on to physical storage as appropriate. WebDAV then defines some basic operations on this model. These operations are invoked physically by an extension of the HTTP protocol, with packets encoded in XML.

According to the WebDAV model, any content store consists of a set of discrete resources arranged in a tree structure. Resources are addressed by hierarchic names (specifically URLs) and are otherwise opaque to the protocol. This means they can represent any items of content, including binary images and multimedia as well as HTML and XML pages.

Objects can, however, be accompanied by metadata which is visible to the protocol. This takes the form of a set of named properties and can be used to hold authorship, subject, versioning, approval status and other such information needed for management and targeting — the specific set relevant to a particular object is indicated by (URL) reference to a schema, though the latter is not included explicitly in the protocol. Properties can also be applied to interior nodes ("folders") as well as to leaf nodes ("files"). Note that binary objects such as images are expected to support metadata, as well as more easily "taggable" formats like XML and HTML. Abstract hyperlinks can also be defined between arbitrary objects and are, likewise, held externally to the objects themselves (at least logically), thereby easing their management.

Against this model a set of new HTTP methods are defined. Each can operate on a single object or on an entire subtree rooted on an object, down to a selectable depth. Such operations on multiple objects are atomic. Methods include:

- PROPFIND: used to return selected or all properties from an object or set of objects, and (implicitly) to list descendent objects
- PROPPATCH: used to perform atomic update and deletion of one or more properties of one or more objects
- LOCK: used to protect objects from colliding updates in a multi-authoring environment
- DELETE: used for atomic removal of one or more objects
- MKCOL: used to create new folders
- COPY: used for atomic replication of sets of objects
- MOVE: corresponding to atomic replication and deletion of objects.

In addition the usual HTTP methods — GET, PUT, POST — are used to read or write actual object contents.

WebDAV itself lacks any facility for searching content stores, whether by properties

(metadata) or by body text. These are the subjects of a sister proposal known as DASL (DAV Search and Location), though this is currently much more fluid than WebDAV. In this, and other such areas, *Athens* has extended the basic WebDAV model, but with the overall intention of adopting relevant standards when they become firm.

Athens Content Storage

As with any content management system, Content Storage is at the heart of *Athens*, so this subsystem will be examined in detail from two different perspectives.

1. The logical perspective, offering the ability to model a wide rage of content and application structures, whether documents, tables, or more general objects.

2. The physical perspective, offering a wide range of storage media to match any required performance, price, scaling, and data-integration needs.

Logical Content Model

The logical content model of *Athens* is based on that of WebDAV and is illustrated in Figure 7.

Thus, content is arranged in a tree structure of *collections* and *resources*. These suggest the folders and documents familiar from filesystems, and provide the same grouping and hierarchic addressing facilities.

However, WebDAV objects have a richer internal structure than simple files, so they may represent more complex Web documents and may include *metadata*. This is achieved by allowing all objects to be equipped with one or more sets of named *properties*.

Operations on content may be applied to groups of objects, rather than to just a single object.

In addition there are optional features concerned with *resource locking*, intended to help organize the safe multi-user update of content stores. These will be implemented in a future *Athens* release. *Athens* also includes some additional features not currently included in WebDAV, such as versioning and querying.

The basic hierarchic structure of the content store provides a familiar paradigm, reflecting conventional filesystems and Webserver stores. This is exploited by the "Explorer"-style folding-tree view presented by the *Athens* User Interface. It provides a convenient basis for addressing and mapping on to URL structures.

The hierarchy can be used to model various real-world relationships, grouping content, for example, according to ownership by individual administrators, or by Website, publication, or page.

Each resource has its own internal structure, consisting of the content body itself (analogous to a file content) and an arbitrary number of properties. A resource can therefore model the following.

1. A *simple file*, where all the content is in the body, and is treated as just an unstructured row of bytes or text characters. There may be some fixed properties, such as content length and modification date, corresponding to those of an ordinary file.

2. A *document* together with its *metadata*; i.e., information about the document such as its author, approval status, subject matter, default publishing template and so on.

3. A *Fielded Database Record*, where all the data is held in the properties, here having the role of database fields.

4. Combinations of the above, e.g., a fielded database record with associated metadata.

Properties have names and types. The particular set can vary from object to object, and may be defined ad hoc or may be dictated by the object's *class*. The latter is part of the *content schema* described later under *user interface*.

Properties may be further organized into *property sheets*, so that name clashes between standard properties and those assigned by different groups of individuals are avoided. They correspond directly to WebDAV's XML namespaces.

Any content field, whether body content or property, may be text (e.g. HTML, or XML), or binary (e.g. images).

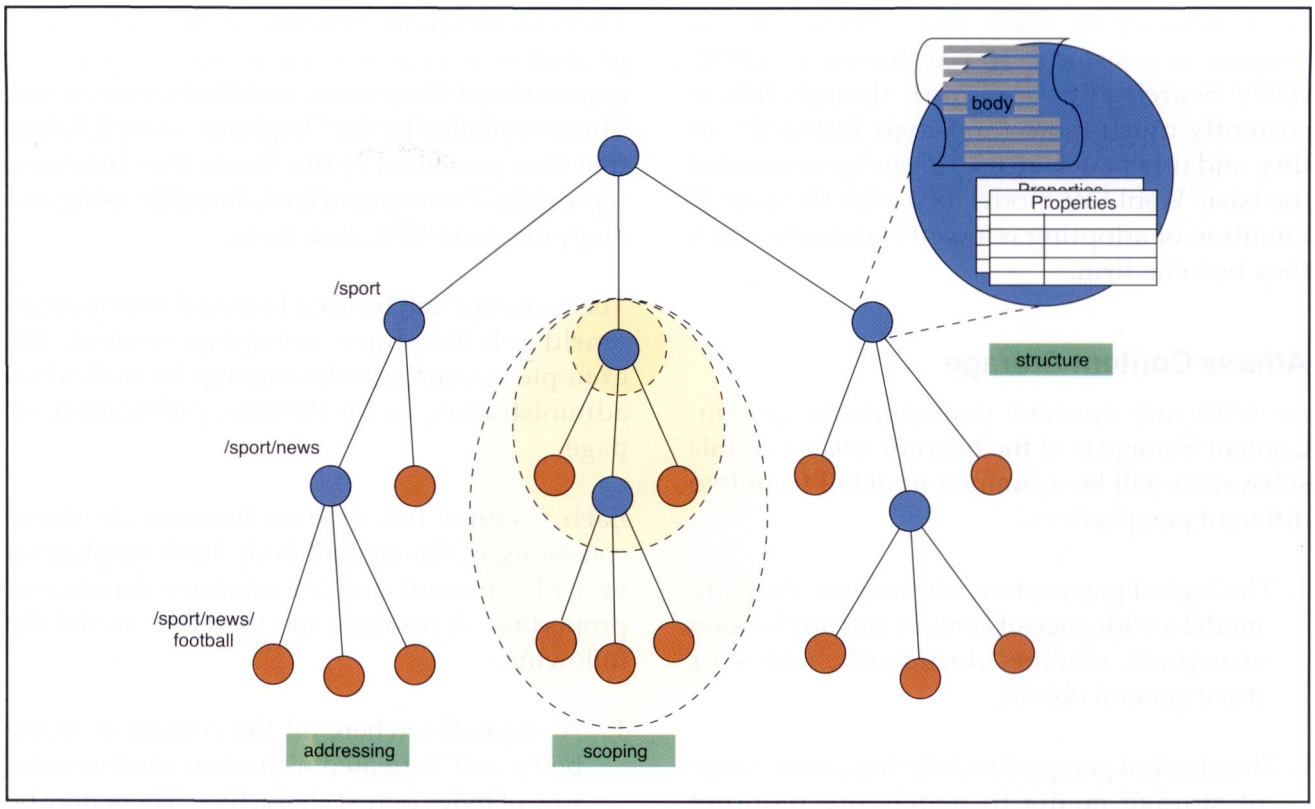


Figure 7: *Athens* **Logical Content Model**

Logical Content Primitives

Operations on items in the logical content model above are also inspired by those of WebDAV.

WebDAV commands are expressed in terms of operations on resources and their components; i.e., body content and properties. HTTP itself provides methods for reading and writing body content. The WebDAV protocol extends HTTP to include methods for reading and writing properties, creating collections (similar to file folders), copying and pasting branches of the hierarchy, etc..

There are also some optional methods, such as those for locking resources over extended periods (e.g., while a document is being revised).

WebDAV command bodies are expressed in XML. For efficiency, Athens uses a Java API (DAV-API) for content access within a single Java Virtual Machine, with object hierarchies replacing XML structures, but the overall command and data semantics remain essentially those of WebDAV.

Commands may be targeted at individual resources (objects), or else at subtrees of resources — for example, it is possible to read a set of properties from a single resource, from a resource and its immediate children, or from the entire subtree rooted on the resource. This improves efficiency by reducing the number of messages. It may also provide powerful macro updates in the *user interface* (not available in Release 1.0).

Future standardization is expected to cover queries based on attributes ("DASL") and remote invocation of executables (e.g. XML RPC). In the shorter term, Athens includes its own support for these topics (Resource Filtering and DAVlets respectively).

Physical Content Storage

Turning now to the physical implementation of the logical content model described above, we recognise that no single conventional mechanism is likely to satisfy every need. Therefore Athens supports a variety of storage media.

1. First is the native Athens storage mechanism intended to satisfy most requirements - the "Virtual File System", or VFS. This is a faithful implementation of the DAV model, implemented in a RDBMS (Oracle or MS SQL Server).

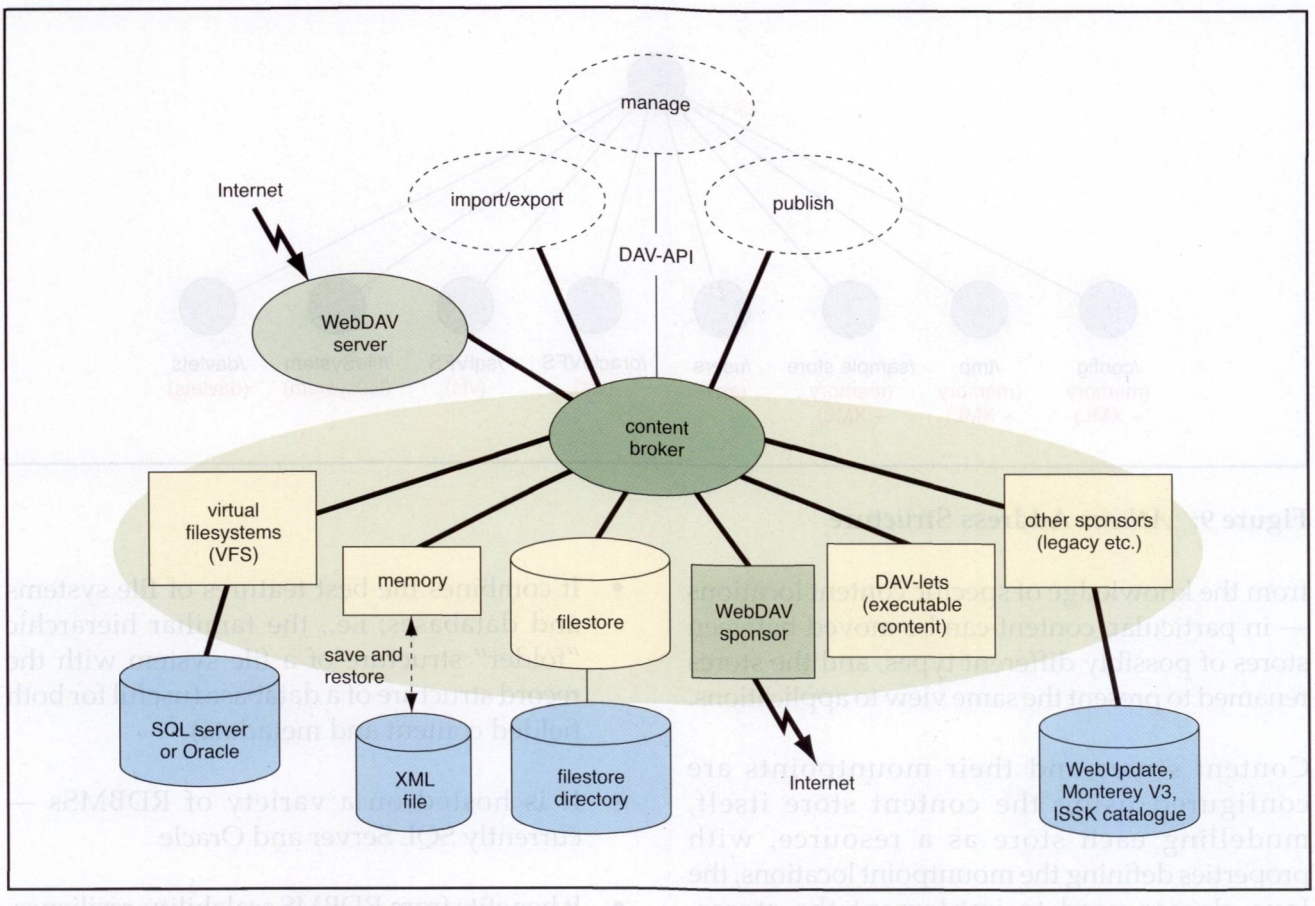


Figure 8: *Athens* Physical Content Storage

2. Then there are transient, memory-based stores, with optional backup in persistent XML files. These are often used for holding configuration parameters.

3. Mappings exist onto real filestore, though with limited features (e.g., a fixed property set). Future versions will allow arbitrary properties to be assigned, as with the VFS.

4. A sponsor allows any remote WebDAV server to be treated as if it were a local content store.

5. A mechanism is provided for allowing content to be generated dynamically and transparently by Java routines (*DAVlets*), as one way of incorporating bespoke functions or business logic.

6. Sponsors may be produced for any form of medium, or for any external or legacy data, which can be moulded into the WebDAV structure.

Any or all of these mechanisms may be used in any solution. The DAV-API interface includes a "content broker" which determines from a content address which physical storage mechanism holds a particular piece of content.

Finally, a WebDAV server gateway allows access to the entire physical storage aggregate by any remote WebDAV client with suitable permission.

The physical content store structure is illustrated in Figure 8.

Address Structure

There is a simple mapping of the physical storage systems described above into the single, logical address space defined by the DAV interface. This is similar to mounting UNIX distributed file systems. Here, however, all mountpoints must exist immediately below the root of the entire content tree, as shown in Figure 9.

Content stores of different types may be used together and multiple instances of each type may be included. Logical names may be used, rather than the names of the physical storage mechanisms, so DAV clients can be isolated from

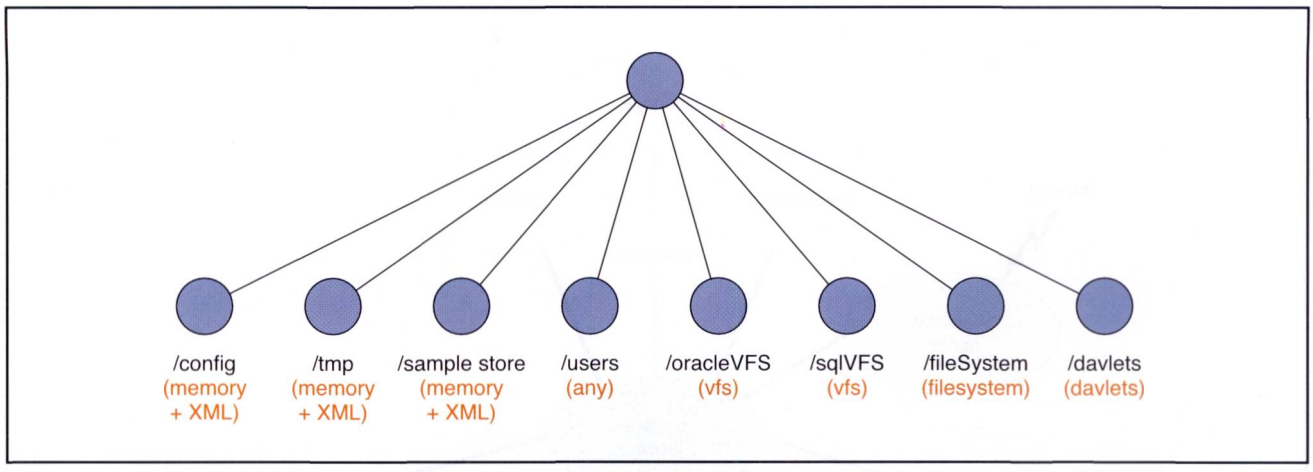


Figure 9: *Athens* Address Structure

from the knowledge of specific content locations — in particular, content can be moved between stores of possibly different types, and the stores renamed to present the same view to applications.

Content stores and their mountpoints are configured using the content store itself, modelling each store as a resource, with properties defining the mountpoint locations, the Java classes used to implement the stores, implementation-specific parameters such as physical directories or database logins, and so on.

The "root" of the logical content store is conventionally mapped onto a memory store which is loaded automatically from an XML file at start-up. This serves as a configuration file (e.g., for the content stores, as above), but allows configuration changes to be applied using the standard *Athens* content management tools.

The Athens Virtual Filesystem (VFS)

Although *Athens* may use a variety of physical content stores for different purposes, as seen above, one is selected as being "a good all rounder" for general use. This is hosted in a relational database (MS SQL Server or Oracle), but implements the full WebDAV file hierarchy, so it is known as the "Virtual File System" or VFS.

The VFS has the following useful features:

- It provides persistent content storage
- It is fully WebDAV compliant (e.g., it supports both text and binary, and may hold an arbitrary set of properties for each item of content)
- It combines the best features of file systems and databases; i.e., the familiar hierarchic "folder" structure of a file system with the record structure of a database (useful for both fielded content and metadata)
- It is hosted on a variety of RDBMSs — currently SQL Server and Oracle
- It benefits from RDBMS scalability, resilience, and administration features
- It has fast, indexed data access
- It has built-in RDBMS schemata — no need for complex database design or associated skills
- Advanced Features include *Temporal Versioning* and full Unicode support.

Active Shortcuts

In addition to the basic WebDAV-derived hierarchic and tabular structures which may be applied to content, *Athens* provides some extra features designed to help model real-world relationships. The first of these is known as an *active shortcut*, which combines the features of a filesystem shortcut or symbolic link, with a database *view*.

A *shortcut* is an item of content whose body contains the address of another item of content. When the shortcut is referenced, the target object is returned automatically and transparently. This means that the same item of content may appear in many locations in the content store, via shortcuts, without needing a separate copy.

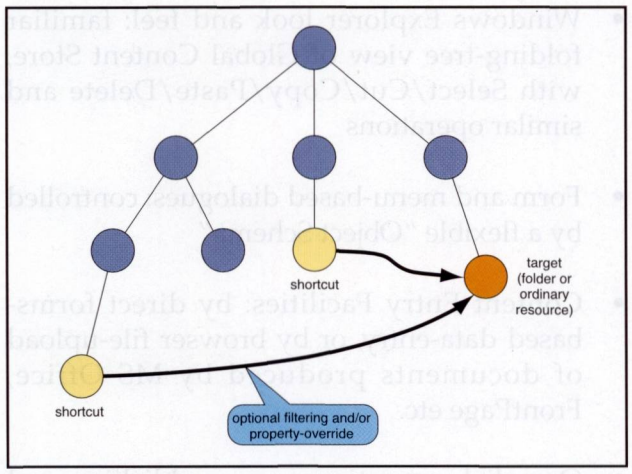


Figure 10: *Shortcuts*

Moreover, if the target item is modified, then the changes are immediately reflected in all the references (*shortcuts*). So, for example, if the target item is a URL to an external site, and that site moves, then only one item needs to be updated instead of searching the entire site looking for references.

Athens shortcuts, however, can do rather more than this. Being objects in their own right, they may have properties. In the case of shortcuts, these may override the properties of the target object. In particular, if the target is a folder with a filter property, then the shortcut may use an alternative filter. Thus, it is possible to define a shortcut to a folder which references only items within the folder satisfying certain conditions, say, only those news items about football. Alternatively, the shortcut may override the template property so that the object looks different when viewed through the *shortcut*.

Use of shortcuts is discussed further, under "Publishing". The basic concept is illustrated in Figure 10.

Modelling Content Relationships

Frequently we wish to represent, not just content, but relationships between content. For example, these may be needed to model:

- The ownership of content by the same author or administrator
- The presence within the same Web site, Web page, or index
- Hyperlinks between pages.

Athens provides a number of ways of representing relationships between items of content. These are illustrated in Figure 11, where three types of relationship are shown.

(1) First, there is the ownership relationship, which usually represents containment of

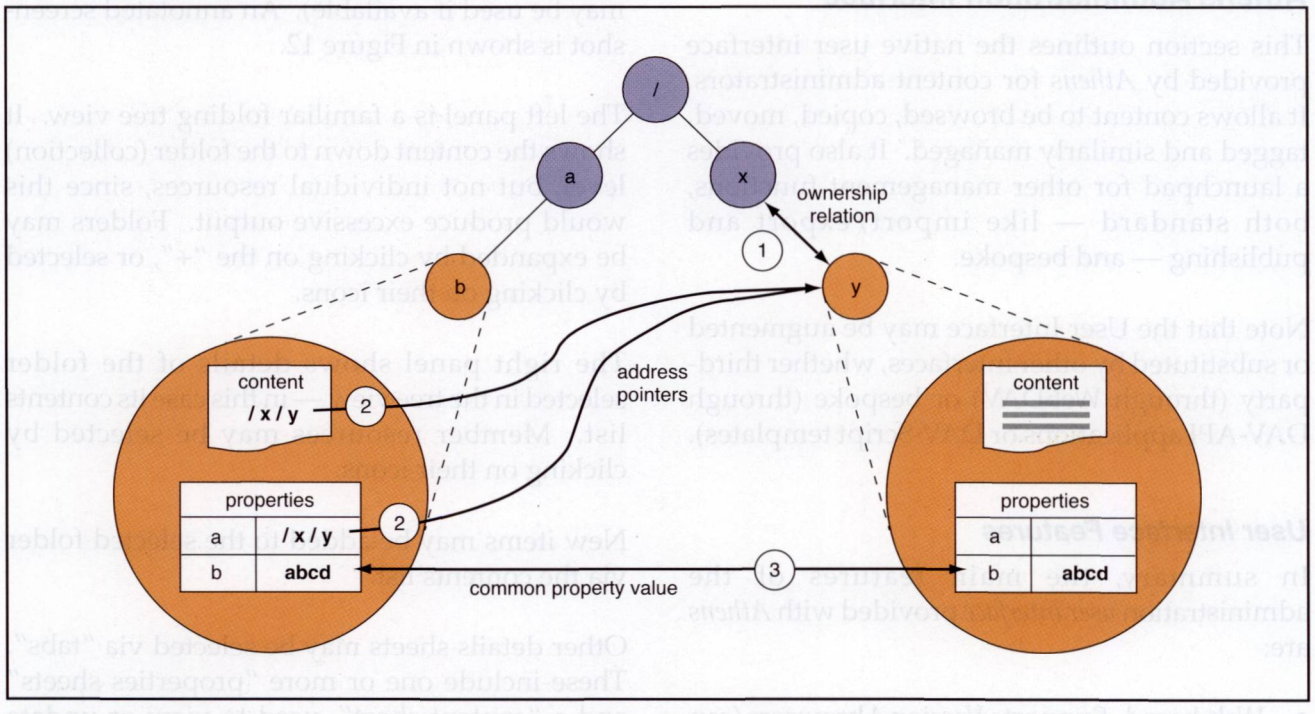


Figure 11: Relationships between items of content

some kind, e.g., pages within publications within sites, or documents within folders owned by individual administrators.

(2) Cross-linking between resources may be achieved by storing the address of the target within either the body content or a property of the source. Shortcuts are a special case. Links may also refer to external resources via URLs. Such links (internal or external) allow an address or URL to be captured in one place, so that it can be changed easily — all other objects and templates refer to the target indirectly via the link object. Note, however, that there is no automatic protection against the address or URL becoming invalid.

(3) True relational "joins" may be achieved by storing the same values in the corresponding properties of related objects — for example, documents may be related by having the same subject or author property. Unlike explicit links (2 above), such relationships have the advantage of adjusting automatically as objects come and go.

In some cases it may be appropriate to devote an entire object to a relationship, its content listing the related objects and its properties describing the relationship.

Athens Administration Interface

This section outlines the native user interface provided by *Athens* for content administrators. It allows content to be browsed, copied, moved, tagged and similarly managed. It also provides a launchpad for other management functions, both standard — like import/export and publishing — and bespoke.

Note that the User Interface may be augmented or substituted by other interfaces, whether third-party (through WebDAV) or bespoke (through DAV-API applications or DAV-Script templates).

User Interface Features

In summary, the main features of the administration *user interface* provided with *Athens* are:

- Web-based. Supports Version 4 browsers (can also exploit IE5 facilities)
- Windows Explorer look and feel: familiar folding-tree view of Global Content Store, with Select/Cut/Copy/Paste/Delete and similar operations
- Form and menu-based dialogues: controlled by a flexible "Object Schema"
- Content Entry Facilities: by direct forms-based data-entry, or by browser file-upload of documents produced by MS Office, FrontPage etc.
- One-click operations: e.g., publishing and XML import/export
- User-dependent views: of Content Tree and User Interface forms
- Highly configurable: can customise toolbars, framesets, templates etc., through content store User Interface model
- Alternative WebDAV clients.

These features will be explored in the following subsection.

Screen Shots

The Native User Interface has a "Windows Explorer" look and feel, which well suits the tree organization of the content store. It is Web-based, needing just a V4 browser (although V5 features may be used if available). An annotated screen-shot is shown in Figure 12.

The left panel is a familiar folding tree view. It shows the content down to the folder (collection) level, but not individual resources, since this would produce excessive output. Folders may be expanded by clicking on the "+", or selected by clicking on their icons.

The right panel shows details of the folder selected in the tree view — in this case its contents list. Member resources may be selected by clicking on their icons.

New items may be added to the selected folder via the contents list.

Other details sheets may be selected via "tabs". These include one or more "properties sheets" and a "content sheet", used to view or update properties and content respectively.

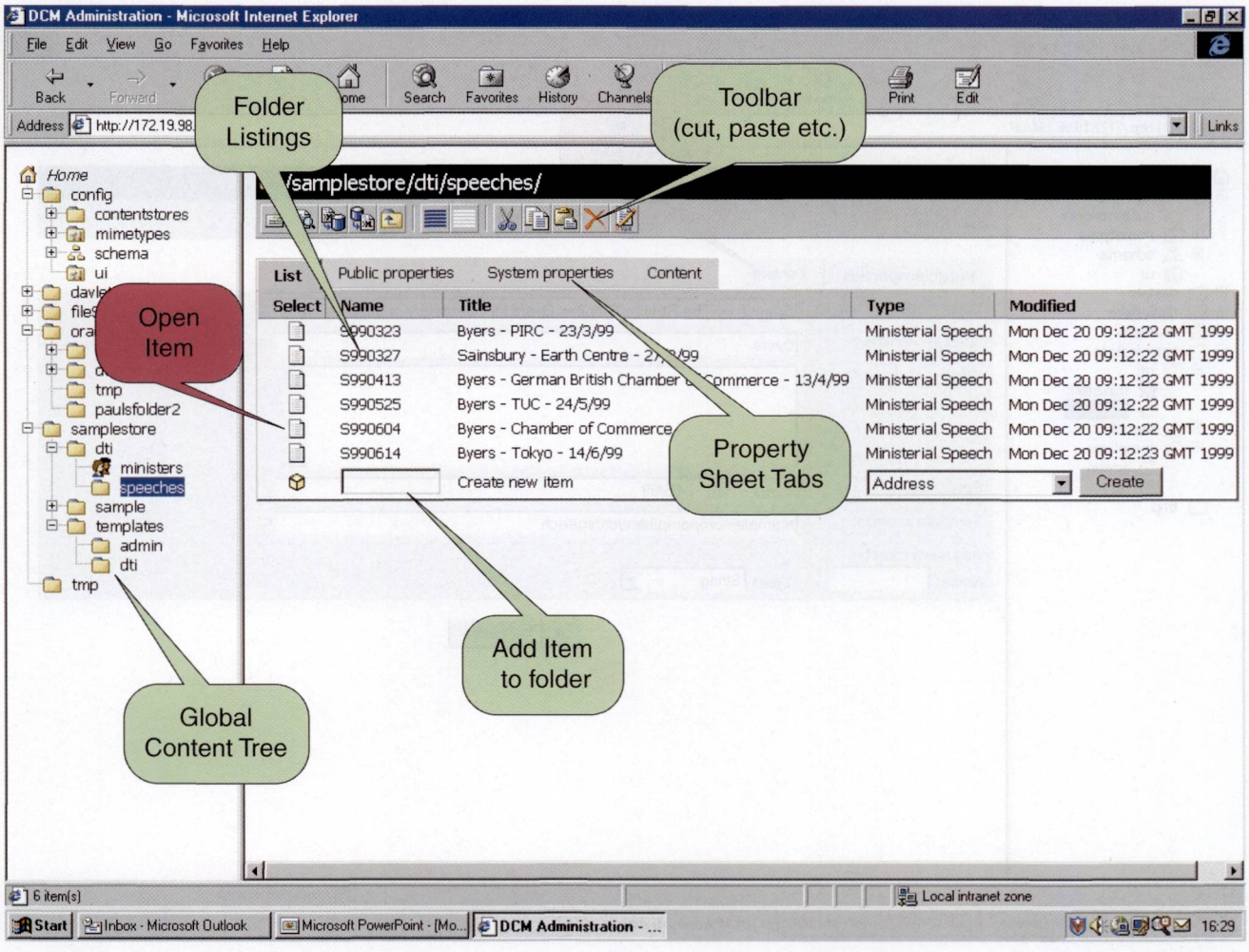


Figure 12: Annotated Screen-shot — top level

The toolbar allows miscellaneous operations to be performed on the selected object. Standard operations include "cut", "copy", "paste", "preview", "select all", "import/export" etc.. In addition, custom buttons may be included, causing templates or Java routines to be executed. The toolbar is fully configurable, including its contents, icons and order.

Clicking on a resource name opens that resource; i.e., it displays the resource's own contents list (if it is a collection), or one of its properties sheets (otherwise). A sample property-sheet screenshot is shown in Figure 13.

Here the resource is not a collection, so its properties are displayed.

There may be other property sheets represented by further tabs. Each corresponds to a *proplist* in the DAV-API, or to an XML *Namespace* in the WebDAV protocol. Property sheets allow properties to be grouped according to purpose, or according to who defined them. They avoid name clashes. In the case shown, there is just one property sheet.

Note that the toolbar has been revised according to the different operations available.

Another tab enables the content sheet to be displayed, of which a sample is shown in Figure 14.

The "content sheet" shows the resource's content and allows it to be updated.

If content is in text format, such as HTML or ASCII, then it is shown directly in a text box, and may be edited (subject to access permission). If it is binary (e.g., an image or a word document) then an option to view it through a suitable viewer application is presented.

Alternatively, the content may be prepared in the browser PC, using a suitable application like

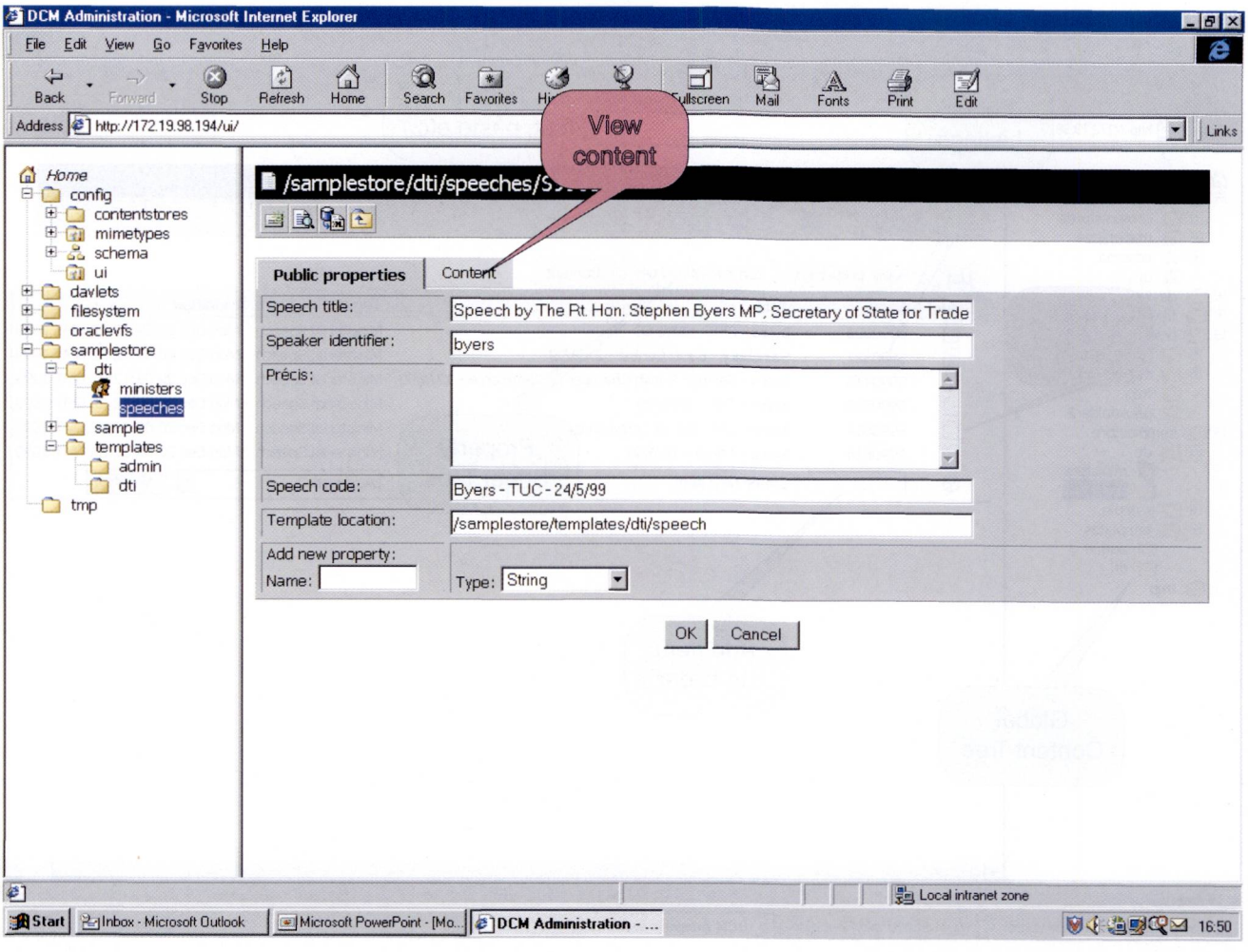


Figure 13: Property Sheet Screen-shot

Front Page or Word, and uploaded by typing in the filename or pressing the browse button.

Resource Schema

The *user interface* derives much benefit from inclusion of a *Resource Schema*.

Although any resource can, in principle, have any set of properties, and all resources might be different in this respect, it is usually best to define a relatively small number of basic object "shapes", or *classes*.

Then, when creating a new object, it is only necessary to state what class it is and what properties it takes are immediately known. By describing such classes to *Athens* in advance, it will automatically generate HTML forms with appropriate fields, names, default values, drop-down menus, help text etc.. This is a major usability benefit and helps to ensure consistency between resources of the same type.

Similarly, in principle a collection may own any type of resource but, in practice, it is often beneficial to restrict them to certain *classes*. Thus, when creating new members, *Athens* will automatically present a drop-down menu of the relevant *classes*.

The set of classes, collectively known as the schema, is represented in the content store and can be configured using the *user interface* itself. Alternatively, through the import/export mechanism, *classes* may be manipulated or exchanged using an XML representation. A basic set of *classes* is provided "out-of-the-box", and these may be refined by the object-oriented "multiple inheritance" feature.

Being implemented above the content stores themselves, *classes* can be used anywhere and altered at will. There is no need to do any custom relational database *schema* design. In summary, the schema:

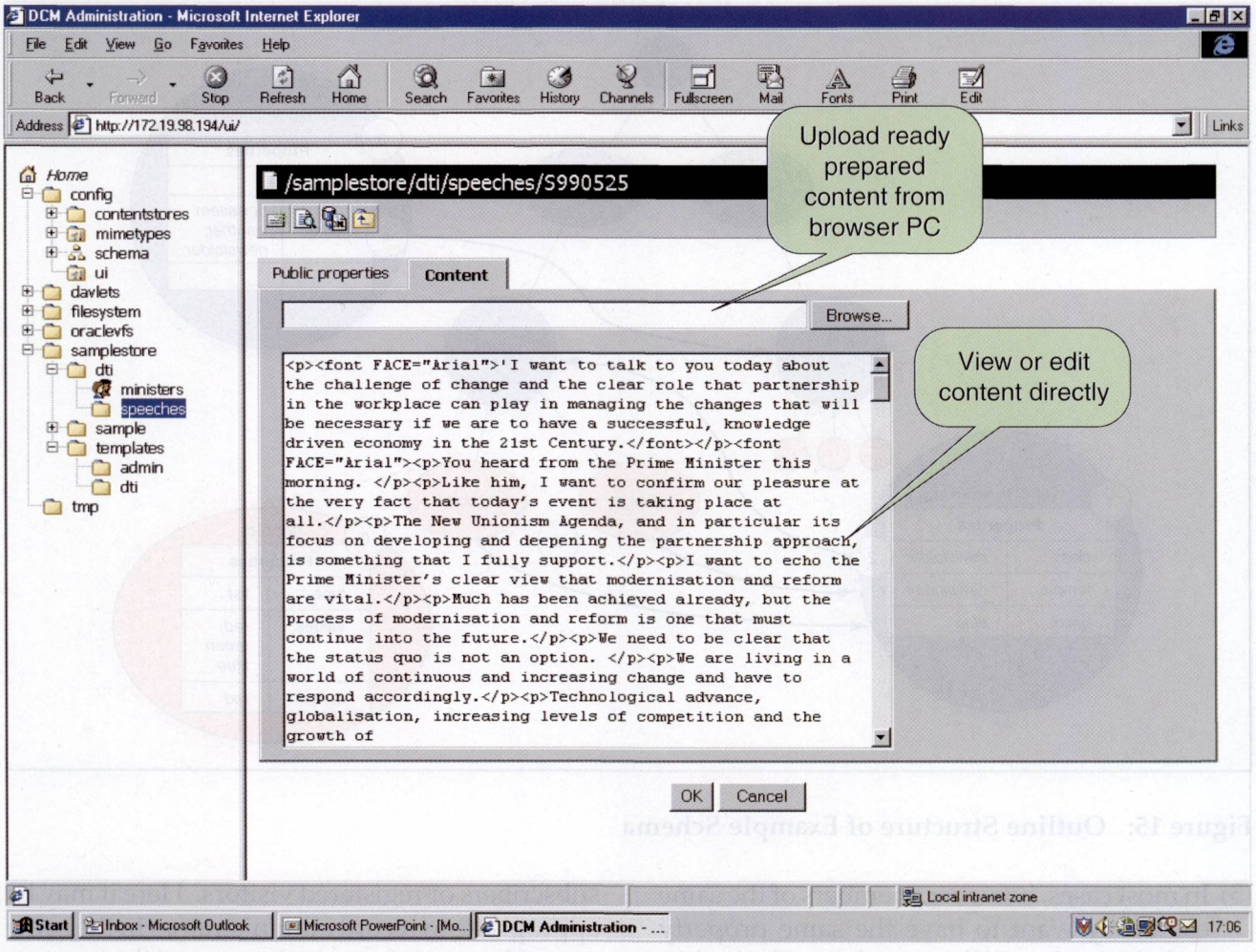


Figure 14: Content Sheet Screen-shot

- Defines the Properties of Objects — names, types, permitted values, defaults
- Furnishes forms and drop-down menus used by the *user interface*
- Controls object placement in the Content Tree — i.e., what classes of object can go where
- Is represented in the Content Tree itself — so it can be configured via the Admin. User Interface, and can import/export in XML format, etc.
- Supports multiple class-inheritance
- Operates above and across all Content Stores.

The schema is made up of classes, each class describing one shape of resource. The outline structure of a class in the content store, and its relation to the objects it describes, is illustrated in Figure 15.

Referring to the diagram, the case of a collection resource (folder) used to hold news items about sport will be considered.

(1) Like all resources, the folder has a set of properties.

(2) One special property declares the class of the resource — in this case it is a newsfolder. The class is actually a reference to another resource in part of the content store used to hold the schema. This second resource describes the attributes of newsfolders via its own properties and children.

(3) First its properties are examined. These can be used to define attribute values shared by all members of the class.

(4) For example, one property might define what types of child resources a newsfolder may have. Here we allow news items, weather reports, and other newsfolders for substructuring.

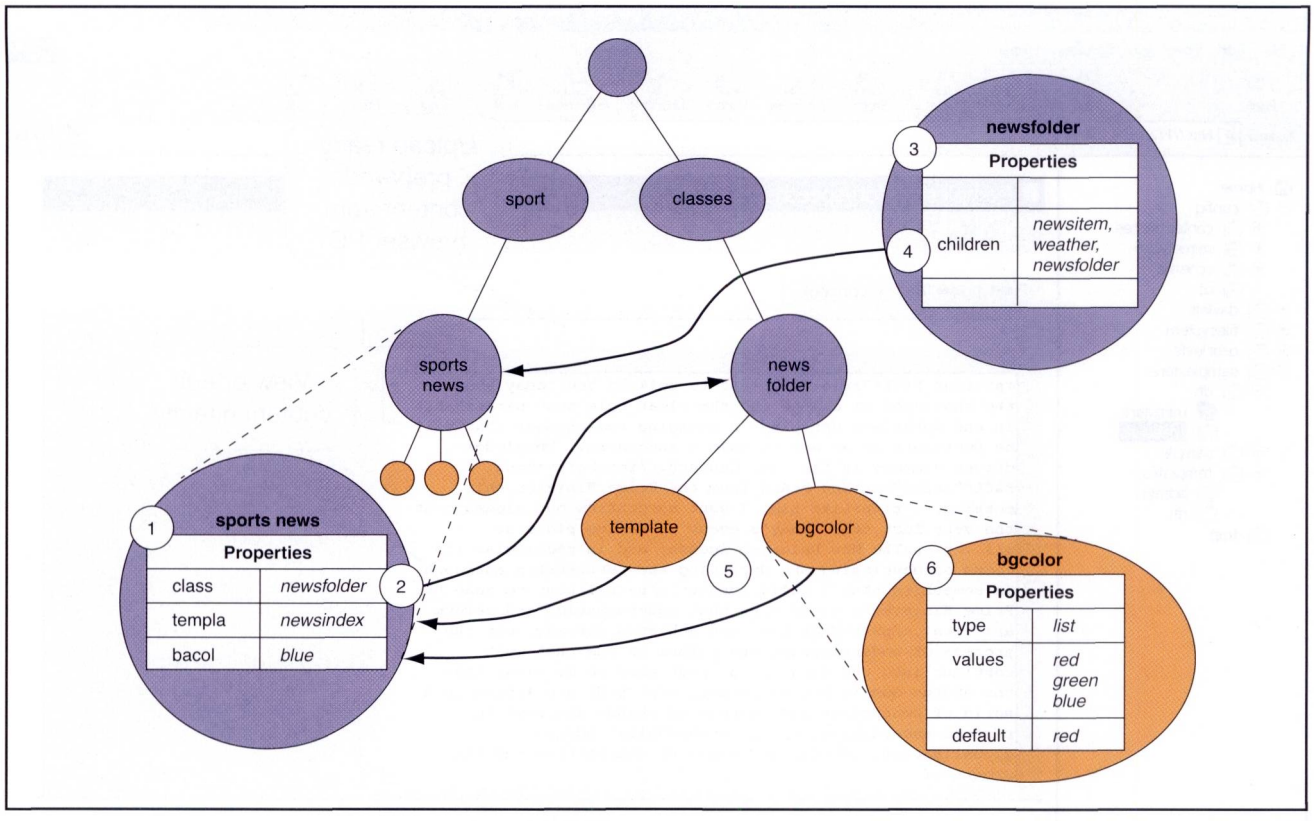


Figure 15: Outline Structure of Example Schema

(5) In most cases, however, members of the same class may want to have the same property definitions, but different values. To do this, each property is represented by a whole resource owned by the class — here we show a template, used to display the list of news items, and a background colour for use in the template.

(6) The reason a whole object is used is so that the attributes of each property can be described, by the properties of the child object. Thus, here we see that the background colour property for newsfolders may be red, blue or green, with red as the default. Actual values are, of course, stored with the object instances.

Managing Administrators

Athens recognizes that a single content management and production system may need to be accessed by many individual administrators, with different skills, preferences and privileges. These need to be recorded and acted upon.

Many solutions will require specific membership databases to be integrated as part of their business application, for example, a directory of shoppers, subscribers or registered visitors. Here it may be appropriate to record administrators in the same way. However, for completeness, *Athens* comes with a simple membership database "out of the box". Either way, members are mapped into the single virtual content store and can be used identically to content.

Typical properties associated with (admin.) users relate to authentication, access control, and preferences. Authentication takes the form of a password, optionally one-way-encrypted.

Access control operates via a "capability list" — in this case a list of content store subtrees which can be accessed, together with the privileges enjoyed on those subtrees. The mechanism avoids needing to store access controls in content stores themselves (some may have a fixed, predefined structure). Privileges are fine-grained, relating to (e.g.) read, write, create, delete, execute, view operations and separately to properties and content. "View" privilege controls whether content branches actually appear in the Content Explorer User Interface. Custom privileges may also be defined. DAV-API and DAV-Script templating each provide methods for checking current user privileges.

In summary:

- Administrator database is held in virtual Content Store — therefore "content classes" can implement "user roles"
- Access-control via "capability lists" attached to users, containing accessible subtrees and corresponding privileges — therefore works on all content stores without modification
- Fine-grained privilege — Read/Write/Execute/Create/Delete etc., plus user-definable, applying separately to body content and properties
- Access checking via Java API or DAV-Script
- "Visibility" privilege influences Admin. User Interface — Content Trees are user-specific
- At Phase 1, all users controlled by a single "Super User" (only this SuperUser has access to the Admin. Database).

Custom Administration Screens

Like many content management systems, *Athens* includes a powerful template language (DAV-Script) as part of its publishing facilities. However, for a suitably privileged user, DAV-Script may modify the content store as well as reading it. Combining this with HTML forms facilities and access to URL parameters makes it possible to implement bespoke administration interfaces in much the same way as end-user Websites.

Such bespoke pages may form freestanding sites, say for different classes of administrator, and/or they may be invoked from the standard Content Explorer interface, e.g., via custom toolbar buttons.

Batch administration

The ability to script administration via DAV-Script lends itself to the possibility of automating administration functions, as in the manner of UNIX Shell scripts. This may be achieved with the present version as a custom feature. However, in a subsequent release of Athens, a Batch Processor application is anticipated, which will execute DAV-Script sequences according to a time schedule, configured through the Content Store.

Content Import and Export

We have already seen two ways of getting raw content into the Content Store — direct data entry through HTML forms, and file-upload from the Web browser PC. Both these are implemented by the Administration User Interface. Likewise, corresponding view and download facilities exist to get raw content back out (for export or modification).

However, sometimes it is necessary to exchange raw content in bulk, and/or with external systems, and/or with some degree of automation. Various import/export facilities are therefore provided.

XML Import/Export

First, an XML-encoded bulk transfer format is defined, with associated Java utility routines. Any subtree may be exported to a file and later imported, maybe into a different system and with a different root node position in the content store. Nested nodes correspond to nested elements within the XML. Properties also correspond to elements, rather than XML attributes, to avoid limitations on their contents which would otherwise occur. Standard tools, such as XSL or the DOM, may be used to convert to or from other XML dialects or external formats (CSVs etc.).

Text content is protected by XML "CDATA" directives, with a mechanism to protect real CDATAs within the content. Binary content is encoded using Base64 or QuotedPrintable as appropriate.

The import/export routines may be executed directly from the Administration User Interface. This may initiate export to a browser window or import from a browser-based file.

Dynamic Access to External Websites

For more dynamic import/export of raw content, without making a physical copy, it is possible to map external sites directly into the local Virtual Content Store address space. Logically, these can be treated as any other (local) content. Physically they are accessed via WebDAV or HTTP over an Intranet or the Internet.

Publishing Raw Content

Publishing, as described in the next section, is usually associated with generating and

delivering documents for end-users. However, the template-based production process may also be used to generate arbitrary text files, including Character Separated Values (CSVs), and XML files conforming to external formats.

Automation of Import/Export

Both XML import/export and WebDAV/HTTP access are available through the DAV-Script template language (see later under Publishing). Import/export operations can, therefore, be automated.

Application Programming Interfaces

For more specialized requirements the Java-based DAV-API interface to content stores can be used directly by bespoke import/export and translation programs (a COM version is also anticipated).

Publishing

In addition to its content storage and management facilities, *Athens* includes powerful page construction and publishing features. An *Athens* system may serve as one or more Websites, delivering "static" or "dynamic" pages directly to end users (e.g. by Web or Email), or it may serve as a content production engine, constructing pages for other Websites or e-Business applications.

Publishing includes facilities for template-based page production, personalization, and multi-purposing for different devices. These features are outlined in this section.

Publishing Features Summary

- Powerful DAV-Script template language
- Multi-device publishing based on templates
- "Dynamic" or "Static" page publishing. Also multi-stage generation to minimize run-time overheads
- Versioning and Staging
- Templates held in Content Store
- Highly configurable, via Scripts or Declarative Modelling
- Multi-server pipeline capability, with fan-in/fan-out.

DAV-Script

DAV-Script is, primarily, a powerful template language for publishing. In common with other template languages, a DAV-Script template consists of an output document (typically HTML) with embedded commands used to insert content from the Content Store. In DAV-Script, commands are encoded as XML tags.

Simple requirements are intended to be simple to achieve, making the language usable by "HTMLers" as well as development programmers. For example, to insert the body of a news article, simply add the following tag:

```
<ds:insert content="/sport/news/000216"/>
```

where /sport/news/000216 is the required content store path. The "ds:" prefix on the tag is used to distinguish the tag as belonging to DavScript, and is known as an "XML Namespace".

Content properties can also be directly addressed, by a suffix, as follows:

```
<ds:insert content="/sport/news/
000216:headline"/>
```

This just inserts the headline property associated with the news article.

There is also a more concise form of the insert command, used to insert values into HTML attributes.

In addition to simple content-insertion there are a number of familiar "programming" facilities (loops, conditions, procedures, variables etc.) which may be used to produce very sophisticated and adaptive Web pages — an index page may automatically adapt to the number of documents currently in a target folder, and may optionally filter them according to property values. For example:

```
<ds:forcontent="/sport/news"
filter="this:subject EQ 'football'">
    ...
</ds:for>
```

This construct loops through all the articles in folder/sport/news, selecting only those about football.

Why invent another template language? What is the relationship with, say, XSLT?

The answer is that DAV-Script is very closely modelled on WebDAV constructs and operations. So that anything that can be done with WebDAV can be done from DAV-Script. For example, this includes writing back to the content store where this is appropriate, possibly setting properties on resources down to a certain depth, as part of some administration function. Even the "local variables" of the WebDAV language behave like DAV resources, and may have associated properties .

From the point of view of templating, the function of DAV-Script tends to be aggregation and filtering of content from different sources, local and remote, in a uniform manner. The result may be rendered directly into HTML, and typically is, but it could equally well be output as XML, for ultimate rendering via XSL. In this case, DAV-Script and XSL are complementary. A future version of DAV-Script is expected to include a <ds:xsl> tag to invoke server-side XSL(T) directly within a template.

However, DAV-Script is entirely self-sufficient both as a template language for publishing and a scripting language for automation.

In summary, DAV-Script:

- Supports templating of any text document (HTML, XML, ASCII, Email etc.)
- Is XML-compliant (uses "ds:" namespace)
- Does simple tasks, like content insertion, simply, but also includes powerful programming features (loops, conditions, variables, expressions, procedures, block-structuring)
- Can filter objects by properties
- Is to be equipped with an Animator/Debugger tool
- Will be equipped with a graphical template editor
- Supports Multi-phase rendering, i.e., pre-generate all but final personalization
- Is closely integrated with WebDAV, e.g. in terms of its addressing, scoping rules, remote access capabilities, ability to both read and write content
- Has uniform content access mechanisms (thanks to the DAV interface)
- Handles both mapped or unmapped (ad hoc) external content stores
- Is extensible via custom Java DAVlets.

DAV-Script Uses

DAV-Script is primarily a content aggregation and page construction language. However, as already suggested, it has various other uses. In summary, it can be used:

- As a template-based page rendering mechanism, separating data and business logic from look and feel
- To implement custom administration screens
- To integrate multiple content sources, possibly drawing on both local and remote documents in a single Web page
- To automate administration procedures — for example by scripting Import/Export, or publishing activities
- For simple procedural customization
- To define simple "methods" on content objects
- For defining queries and filters
- To complement XSL(T).

DAVlets

DAV-Script contains a range of built-in facilities for common operations like content access, string processing, and encoding. In addition, custom functions may be added as Java classes, via a special content store. These are known as DAVlets.

DAVlets are addressed like any other content resources. For example, they may be used with the standard DAV-Script `<ds:insert>` command. In this case, when the command is executed, the DAVlet is run and its output used as the content to be inserted into the Web page. Some standard facilities are implemented in this way, for example, the date function, which appears to be an item of content which somehow always has today's date in it.

Figure 16: DAV-Script Template Animator

Similarly, a DAVlet can be used as the target of a `<ds:write>` command. The data being written to the resource (DAVlet) is passed to the DAVlet as its arguments.

Such transparent access to DAVlets is not confined to use by DAV-Script templates. HTPP/WebDAV get and put operations can also target DAVlets as if they were ordinary content.

For those functions which do not readily map on to get/put or insert/write semantics, DAV-Script provides an explicit `<ds:call>` command. For example, this allows data to be both passed as arguments and returned as content, in a single call.

DAV-Script Animator/Debugger

In the near future, it is intended to equip DAV-Script with a set of powerful tools. First, and already demonstrable in prototype, will be a template animator/debugger. This allows a template to be executed command by command, so that the incremental affect on the output screen can be monitored. It is similar to the windows-based interactive debuggers available in some programming environments, except that it is entirely Web-based and requires no more than a Version 4 browser. A screenshot from the prototype is shown above in Figure 16.

The main features of DAV-Script are summarised below.

(1) The Control Window allows an object/template combination to be specified, and controls the step-wise rendering process. A toolbar is provided to allow single step, or "fast forward" (i.e. multiple steps, say five at a time), and restart from the beginning. It is also possible to jump to the end and step backwards. Or a step number can be entered explicitly.

(2) The Template Window shows a portion of the template being rendered. All the DAV-Script

commands are highlighted, and the command which is about to execute is highlighted in a different colour (here red).

(3) Clicking on any of the DAV-Script commands in the Template Window causes a pop-up help window to be displayed for that command.

(4) The View Window shows the page output so far. Rather than just truncate it at the current generation step, the underlying HTML is adjusted to make it well-formed; i.e., to include end tags. This avoids most problems which might arise from passing incomplete HTML sequences to the browser. Through the Control Window, an option is available to view the rendered HTML text instead of the resulting page image.

(5) The output view is seeded with small hyperlinks known as "locators" (here they are represented by little green "L" blobs). There is one such locator for each DAV-Script command instance in the template, and each indicates the location within the output which was generated by the corresponding command instance. The locator's tool tip shows the command instance and the step number. Clicking on the locator will automatically "rewind" the rendering process and step to the selected point. This is useful for quickly finding the command which generated a particular (maybe erroneous) item of output.

(6) Finally there is a Watch Window, in which the current values of selected local variables are displayed. A default set of variable names can be configured, and others can be entered into the Control Window. This feature is useful for debugging more complex scripts.

Dynamic Publishing

Athens can behave as a dynamic Webserver, constructing pages and delivering them directly to end users on demand, in response to page requests. This enables rapidly changing content, or the results of application functions, to be published, and it also enables pages to be personalized or otherwise targeted.

To achieve dynamic publishing, URLs arriving in Web requests are interpreted directly as addresses in the Virtual Content Store (strictly, the first part of the URL addresses the publishing servlet, and the remainder is the content address). The query string, if any, is passed to the rendering process as arguments (made to look like temporary content items).

The selected object may be a template , or may have a template address associated with it (as a property). In the latter case, the template address may be modified by the provisions for multi-device support, or personalization (see later). Once established, the resulting object/template combination is passed to the DAV-Script engine to produce the output page.

Static Publishing

Instead of publishing pages dynamically on demand, batches of pages may be generated in advance and stored either back in content store or in a standard Webserver's filestore (which amount to the same thing if the filestore has been mounted into the virtual content store). This is generally much more efficient for pages whose content is not expected to change frequently nor to be dependent on individual end users.

Static publishing can be initiated by a button in the Administration User Interface, or by a batch DAV-Script, and can apply to an entire subtree. Folders within the (raw) content store can indicate (through their properties) what the corresponding target file folder should be. Typically, static publishing will be repeated weekly, nightly or twice-daily to refresh the content.

Phased Publishing

A hybrid approach to publishing allows pages to be partially rendered as a static publishing exercise, to be finished off dynamically at run time, e.g., for personalization. This helps to minimize the run-time cost of page production on live Websites.

Phased publishing is supported by DAV-Script. Each command may be marked with an attribute indicating the publishing phase at which it should be expanded. Thus, "body material" may be marked as Phase 1, while "Personalized material" may be marked as "Phase 2". If a static publishing process is initiated quoting "Phase 1", then any "Phase 2" (personalization) commands will be skipped, and will remain in the output.

Such interim output will be stored in another part of the virtual content store, not in a Webserver's filestore, ready for dynamic publishing (at Phase 2).

E-Mail Publishing

As a standard feature, rendered pages may be despatched via E-Mail rather than through a Webserver. Simple address lists may be configured through the content store.

Future facilities are expected to include agent-style mailshot processing. At present such facilities can be implemented as bespoke features.

Multi-Device Publishing

To date, most browsers have been hosted on orthodox computers, in particular PCs. All support some version of HTML and HTTP, but extensions to these protocols, often brand-specific, have caused severe problems for page authors. Frequently, therefore, pages are either targeted at a "suitably low common denominator" (say version 3 or Version 4 core features), or are "optimized" for one particular feature level or one particular brand of browser.

Templating provides the ability to make Web pages adaptive to the browser type, which is available to the template as an argument (masquerading as content). DAV-Script's conditional features can be used to vary the HTML output as required.

However, this can make templates quite complex. Furthermore, a growing range of alternative devices, such as mobile phones and messagers, and Web-enabled TV, use protocols divergent from the original HTTP/HTML pair, for example, WAP and WML. In addition, it is not always appropriate to include the same content in all these instances, because of limitations in screen size or bandwidth.

In such cases it is often more appropriate to use separate templates, so one item of content may be rendered in different ways. To support this approach, Athens provides a simple mechanism for choosing the appropriate template in any situation. As mentioned earlier, each object may be equipped with a property value indicating the default template used to render it, as a hierarchic address in the Content Store. It is also possible to configure, via the Content Store, a modification to this address based on the current device type, so that an alternative template may be selected.

It works by allowing an arbitrary string within the default template address to be substituted by an alternative string, according to device type. For example, suppose the default template for a Web page is "/sport/templates/newsindex", but there is a WAP-specific version in "/sport/templates/wap/newsindex" and a digital TV version in "/sport/templates/dtv/newsindex".

Then, configuration entries such as:
would have the required effect.

It is also recognized that simply using alternative templates is not always enough—page navigation and other procedures may need to be different for different devices. One approach is to use separate sites for each major class of device. Athens provides support for this, as illustrated below under Declarative Modelling.

Personalization

One of the most significant features of e-Business is its potential for adapting itself dynamically to the needs or attributes of individual customers. For example, an on-line shop may be organized so that the kind of goods a visitor has previously investigated or bought are at the front of the store.

Device	Search String	Substitution String
wap	/templates/	/templates/wap/
dtv	/templates/	/templates/dtv/

Or an on-line magazine (Webzine) may contain just those articles likely to be of interest to the current visitor. In the limit, every visitor may have his or her own unique view of the business or its content, which would not be possible with a high street shop or a printed magazine.

To achieve appropriate personalization of content there are many technologies and products now available or emerging, ranging from simple keyword relations between users and articles, through free-text search engines used to deduce relationships, to sophisticated neural network systems which can learn a user's behaviour patterns. The task is twofold:

- To establish an individual's needs, interests and "hot buttons"

- To use these attributes to select appropriate content and behaviour.

A follow-up task may be to analyse the results for general marketing insight.

A user's attributes may be obtained explicitly from some form of on-line registration process, possibly linked to some off-line knowledge base, such as a CRM (loyalty card) system or magazine subscription. Alternatively, and especially for anonymous users, they may need to be deduced from the user's behaviour, and/or the behaviour of similar users.

The content-selection task may also be driven explicitly, e.g., by an end-user choosing his or her own organization for a portal homepage. More commonly, however, such selection is left to the system itself, requiring it to match user attributes against content properties.

Athens does not attempt to duplicate the highly specialized technology needed for world-class personalization support, nor does it constrain its adopters to any specific technology or product. Rather, it seeks to provide lower level support for such technologies. For example:

- It can store metadata, particularly subject indications, with any item of content. This can be used both to deduce a visitor's interests (from the subject fields of the content that he or she has requested previously), and to select suitable content for delivery
- It provides keyword search facilities, to locate content based on property values
- It provides for storage of users and their attributes, i.e., in a suitable content store, with properties used to hold attributes, and classes used to represent user types
- It provides for highly parametric site behaviour, any of the parameters being potentially subject to personalization. For instance, personalization might be applied to simple display attributes like background colour and font details, through to complete page layouts via alternative templates.

Nevertheless, it is possible to perform a quite high degree of personalization with *Athens* right "out of the box", using the above features combined with DAV-Script. Within any template, the current user information is available as an item of content, with known user-attributes represented as content properties. It is, therefore, possible to match attributes to the properties of (real) content .

For example, the following might generate a list of news items relating to the current user's interests:

```
<ds:for name="newsitem" content="/sport/
news" filter="newsitem:subject IN
/user:interests">
  ...<!- Generate a link to newsitem -
>...
</ds:for>
```

Here it assumed that the user's interests property is a list of keywords which might otherwise occur in a content item's subject property.

Declarative Modelling — Websites

The structure of a Website published by *Athens* has been shown generally to reflect that of the underlying raw content store. This may not always be appropriate. For example, the raw content might need to be organized according to physical location (if it is distributed), or ownership. Furthermore, the content may be required to populate multiple Websites of different structures and with different selections in each. Keeping multiple copies of content would consume space and incur an administrative and performance overhead to keep them in step.
To solve this kind of problem active shortcuts can be used to build alternative views on the same set of raw content. This is an example of declarative modelling, whereby administrative processes (in this case publishing) are represented by resources and properties in the content store. The advantage of such a model is that symbolic operations on the model automatically result in operations in the real world. So, copying and pasting a shortcut into one or more models causes the referenced content items to be published in the corresponding Websites. This is illustrated in Figure 17.

The stages are as follows.

(1) The content store is divided into three sections, reflected by branches or groups of

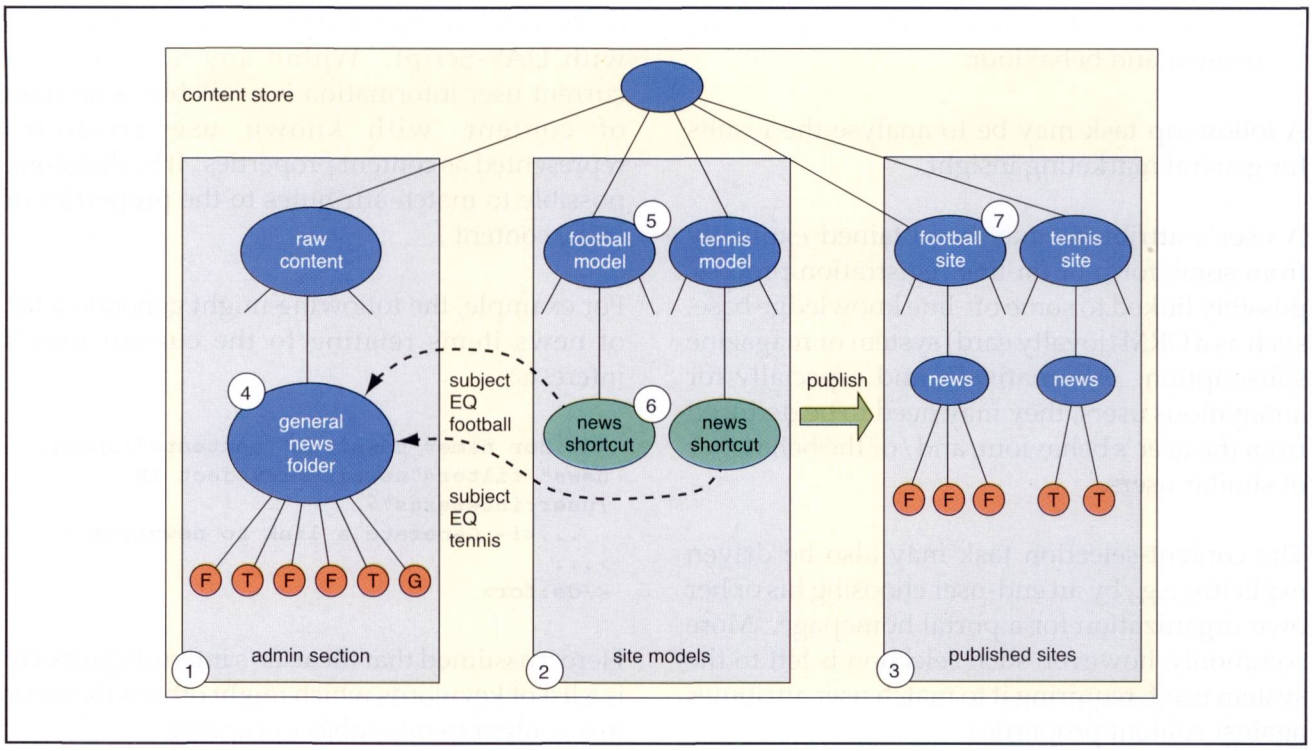


Figure 17: Example of Declarative Modelling — Websites

branches in the content tree. The first section contains raw content for use in all sites.

(2) The second section contains models of the target sites.

(3) The third section contains published sites themselves. For static publishing, this may be Webserver filesystems mapped into the Content Store. For dynamic publishing, the section will be virtual site images as seen by site visitors, the component pages being generated on demand.

(4) The Raw content is arranged for administrative convenience. Here, all news items have been grouped into a single folder — "F" denotes items about football, "T" about tennis, and "G" about golf.

(5) Site models contain appropriate folder structures, templates, brand images, and anything else which might be site-specific. Here we have defined two sites, one about football, the other about tennis.

(6) In most cases, however, actual content will be replaced by shortcuts to items in the raw content store. Here we have included shortcuts to the news folder, with different filters so that the two sites include only appropriate articles.

(7) When publishing occurs (dynamic or static) the shortcuts are followed to retrieve the raw content, but the properties defined in the models are used to apply content filtering, template selection, and other customizations specific to the target sites.

Note that adding a new news item to the raw content folder will automatically add the news item to any site to which it is relevant.

Declarative Modelling — Page Structure

In some cases, complex pages such as portals may also benefit from declarative modelling. In this case, instead of a page being represented by a single object or template, it is described by a subtree of resources, each representing some page component such as a featured news article or site index.

Again, shortcuts are used so that portal components can be changed just by redirecting the shortcut, without needing to change the page itself, or write any new HTML.

This is illustrated in Figure 18 and works as follows.

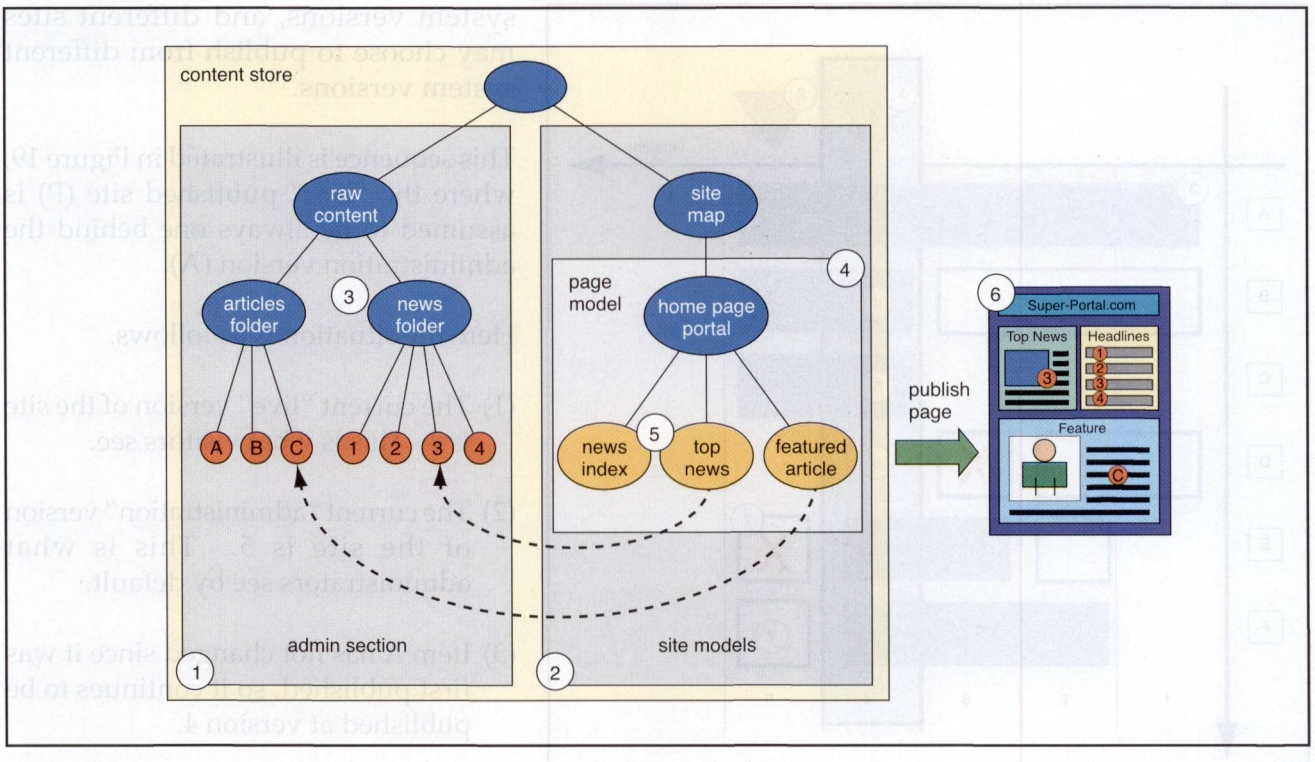


Figure 18: Example of Declarative Modelling — Page Structure

(1) The content store is again divided into sections, one for the raw content...

(2) ...and one for the target site models.

(3) Raw content is again organized for administrative convenience. Here we have two folders, one for editorial articles and one for news items.

(4) Site models are defined as before, but here the internal structure of the home page portal has also been exposed.

(5) Each item in the portal is represented by a shortcut to the folders and items in the raw content section. The portal template will usually generate indexes for folder shortcuts, and summaries for resource shortcuts.

(6) When the site is published, the shortcuts are followed to build the page as desired. Re-routing any shortcut will cause the associated section of the portal to be updated, while adding or removing items to or from a folder will likewise automatically alter indexes.

Temporal Versioning

It has been shown how multiple versions of the same content, targeted at different devices or Websites, may be held together in the same virtual content store. This may extend to other situations, for example different natural languages. Such versions have the property that they are all "valid" at once. This is called *Spatial Versioning* to indicate that such versions are separated across space rather than time.

However, *Athens* also allows concurrent storage of versions separated by *time*. Thus, items of content may be accompanied by newer versions still in preparation, or older versions needed for archive or possible regression; this is called *Temporal Versioning*. Any number of versions of any document may exist concurrently.

The (temporal) versions of a document are distinguished by a simple numeric label (1, 2, 3 ...). To simplify organization, a version number can also be applied to an entire system. This means that, when a particular item of content is requested, the version which corresponds to that of the current system version number is selected. If that version does not exist for this item, then the next *lower* version which does exist is selected.

This means that a new system version can be created without making any new copies of any content — the previous versions will continue to be used in the new system version until they are modified. When an item is modified, a new copy

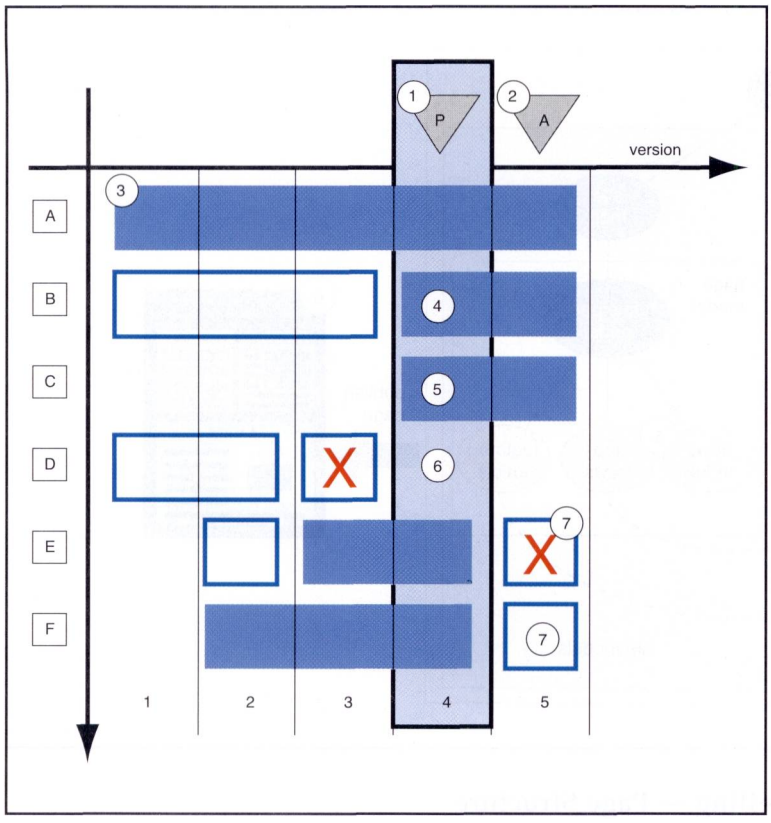


Figure 19: Example of Temporal Versioning

is made automatically and transparently, and assigned the new version number, retaining the original unmodified version with the original version number.

Likewise, if an item is deleted then it is merely marked as deleted at the new version number — the contents are retained in the content store labelled with the previous version.

Within these simple rules, versioning can be used in a number of ways, for both long-term strategic site redesigns or campaigns, and day-to-day updates. A simple approach assigns two current versions — version N for "live" publishing (i.e. what site visitors see), and version N+1 for administering. This means that whatever site administrators do, their effects are not visible on the live sites. At some appropriate time, when the new material is complete and behaves correctly, both the live and administration versions are incremented, so that the previous administration version becomes visible to visitors, and a new administration version is spawned. If necessary, the live site may be regressed back to its previous version. In addition, individual administrators may choose to view different system versions, and different sites may choose to publish from different system versions.

This sequence is illustrated in Figure 19, where the "live" published site (P) is assumed to be always one behind the administration version (A).

Here the situation is as follows.

(1) The current "live" version of the site is 4. This is what visitors see.

(2) The current "administration" version of the site is 5. This is what administrators see by default.

(3) Item A has not changed since it was first published, so it continues to be published at version 4.

(4) Item B has been modified at version 4. The previous version remains in case it becomes necessary to regress the live site.

(5) Item C is a brand new item at version 4. If we regressed the live site to version 3 it would seem to disappear.

(6) Item D used to exist, but was deleted in a previous version, so it no longer appears in the live site. It would reappear if we regressed to version 2.

(7) In the current administration site, items A, B, C, and D remain unchanged, item E is deleted, and item F has been modified. These changes do not yet affect the live site.

Integration & Customization

As a solution kit, *Athens* is required to form the basis of a wide variety of customer solutions, and to integrate with other solution kits. This section outlines the main features provided.

External Interfaces

In order to achieve the desired degree of interworking, *Athens* is equipped with a wide range of open interfaces. The main ones are illustrated in Figure 20.

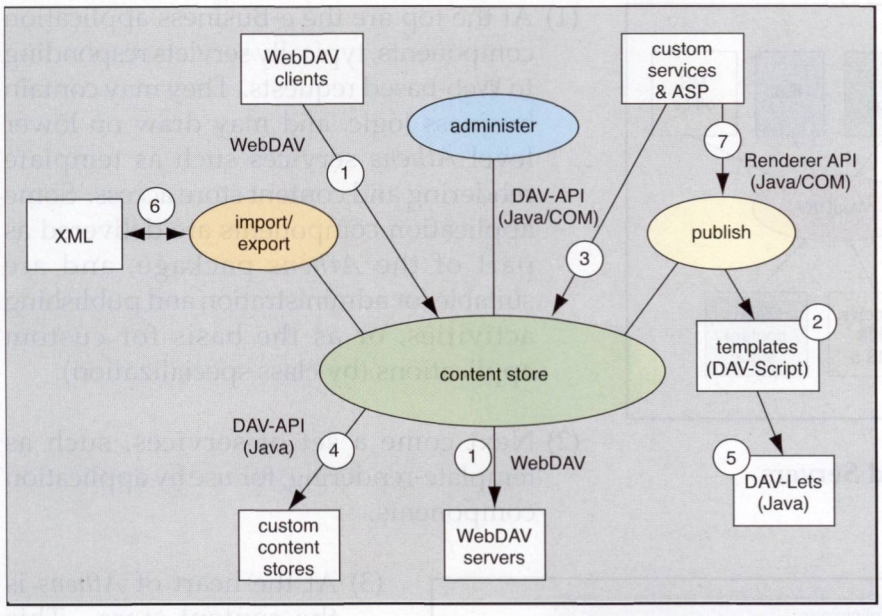


Figure 20: Athens — Primary Integration Interfaces

In summary, these interfaces are as follows.

(1) **WebDAV**: This allows third-party authoring and content management clients to be used in conjunction with *Athens* content stores, and/or allows third party content stores to be mapped into a local Athens content address space.

(2) **DAV-Script**: Templates and scripts written in DAV-Script may be used to customize operation of *Athens* in various ways, for example, by providing bespoke administration interfaces. DAV-Script may also be used to integrate content from remote Websites using its WebDAV and HTTP client facilities.

(3) **DAV-API**: This allows custom Java applications to access *Athens* content stores.

(4) **Bespoke content stores**: DAV-API also allows bespoke content stores to be integrated into *Athens* itself.

(5) **DAVlets**: These allow typically small, discrete Java routines to be added to the DAV-Script language for bespoke purposes, for example dynamic acquisition or preprocessing of content prior to rendering.

(6) **XML Import/Export**: These routines allow bulk content to be exchanged with external sites, in the industry-standard XML format.

(7) **Content Services**: Most of the functionality embodied in the *Athens* subsystems (import/export, Admin. User Interface, and Publishing) are exposed to Java programmers so that new servlets may be written for custom applications. Example services are template-rendering and schema access.

Core *Athens* facilities are programmed in Java, and this, is currently the likely choice for programmatic extensions. However, COM wrappers are being developed in order to enable:

- *Athens* functions to be invoked from COM environments, for example ASP pages and Visual basic
- *Athens* to invoke COM-based functions as, for example, DAVlets.

In addition, forthcoming exploitation of middleware platforms such as EJB will enable open access via COM+, CORBA, and JNI interfaces.

WebDAV Integration

WebDAV offers the loosest form of coupling between *Athens* and external components or applications. It can also allow the components of *Athens* to be distributed. An example of how various clients and services (content stores) may be integrated is shown in Figure 21.

Athens Internal Structure

For closer integration, some knowledge is needed of the internal structure of *Athens* and its interfaces.

From a programming viewpoint, *Athens* has the basic layered internal structure illustrated in Figure 22. The diagram also indicates the main places where custom code might be inserted.

Athens Java code is divided into four layers of functionality as follows.

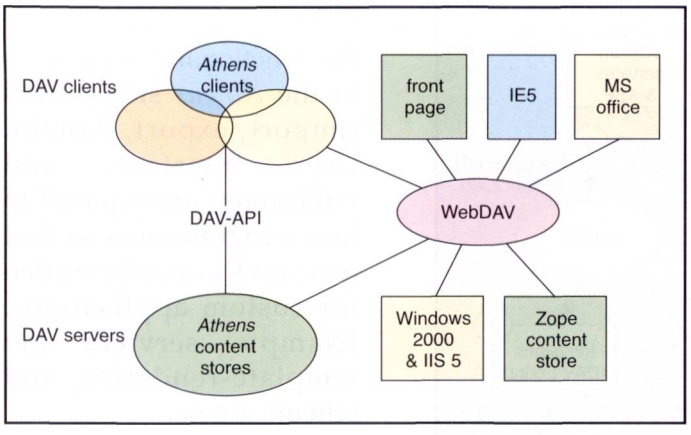


Figure 21: WebDAV Clients and Servers

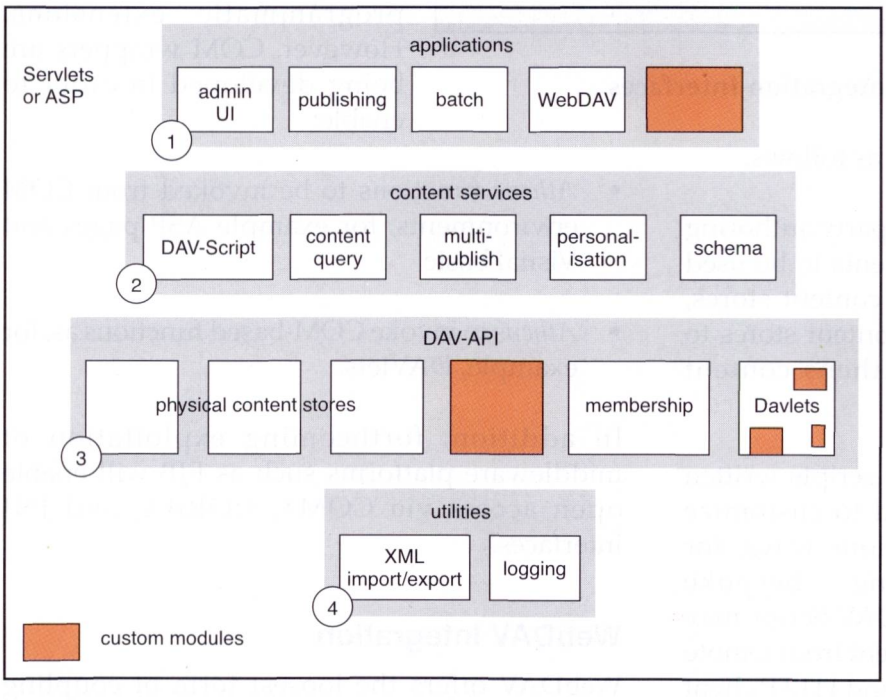


Figure 22: Athens Internal Structure and Interfaces

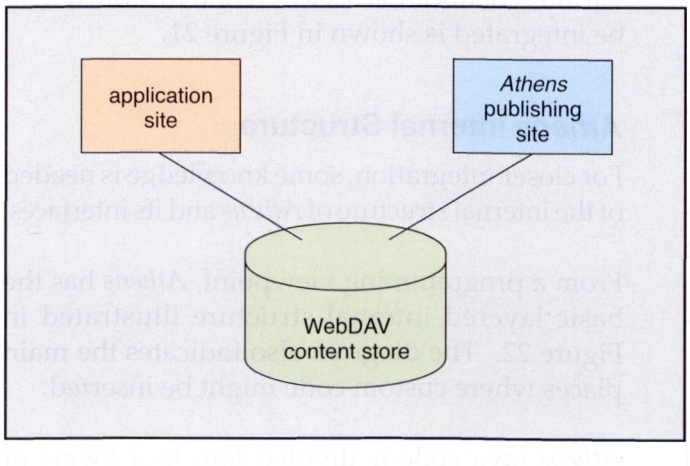


Figure 23: Arms' length WebDAV Integration

(1) At the top are the e-Business application components, typically servlets responding to Web-based requests. They may contain business logic, and may draw on lower level *Athens* services such as template rendering and content store access. Some application components are delivered as part of the *Athens* package, and are suitable for administration and publishing activities, or as the basis for custom applications (by class specialization).

(2) Next come a set of services, such as template-rendering, for use by application components.

(3) At the heart of *Athens* is the content store. This may be customized by adding new stores and/or writing "DAVlets".

(4) Finally there are some utility routines, such as logging, which may be used by any module.

As will be apparent, the first release (1.0) of *Athens* will integrate most readily with Java-based applications. COM wrappers are currently being prepared for the major interfaces so that they may be used, e.g., from Microsoft Active Server Pages and Visual Basic, and so that DAVlets may be developed as COM objects.

Integration Approaches

To write an application based on *Athens*, or to integrate an existing application, there are four main approaches. These are outlined in the following subsections. Of course, they may be combined in various ways.

WebDAV approach

The simplest approach is to integrate at the data level only, without attempting to share program functions. There are two basic ways of doing this. The first uses the industry-standard WebDAV protocol to connect an

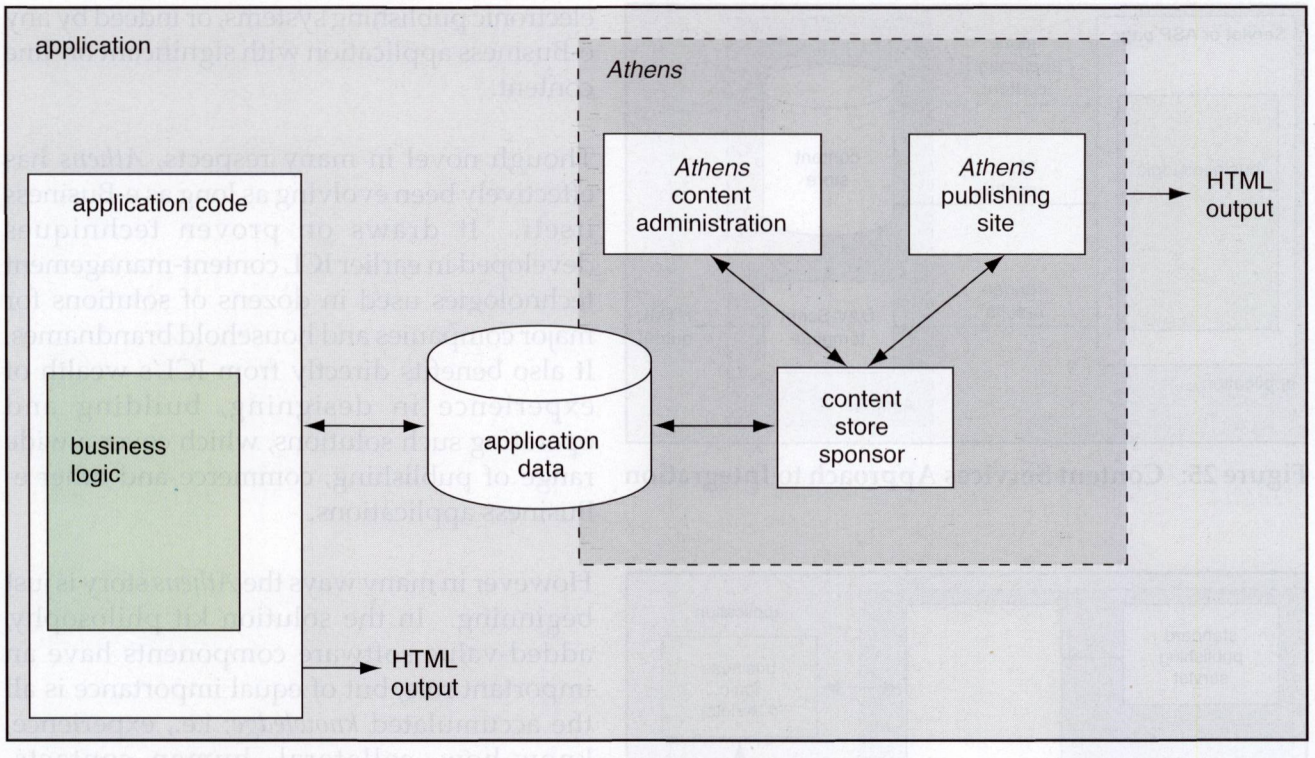


Figure 24: Shared Content-Store Approach to Integration

application to a (remote) *Athens* content store. This allows the application to provide the content for an external *Athens* site to publish, and allows the application to make use of the content administered by *Athens*. But integration is very much at "arms length". It is illustrated in Figure 23.

Shared Content-Store Approach

In some cases the application may have its own specialized data storage needs, with more immediate access than is offered through WebDAV. For example, it may have a "legacy" database. In this case it is often possible to make this data available to *Athens* using a custom Content Store sponsor — *Athens* itself may be running on a different server, connected, for example, by SQL. This means that an application's data can be included in a publishing site, along with other content, without having to call the services of *Athens* directly. It may also be possible to manage aspects the application's data through the *Athens* Administration User Interface. However, it does not exploit the full potential of *Athens*, for example, the ability to render dynamic application results for different users and devices.

The shared-content approach is often the first phase of integration for an existing application, since it avoids mixed-language programming issues. This is illustrated in Figure 24.

Content Services Approach

To take advantage of the content services of *Athens* within an application itself, we have to decide how to split content functions from the application business logic.

The business logic may be included at the application level, within the servlet (or, in future, within an ASP page), and to call *Athens* services as required. Content supporting the application may be held in *Athens* Content Store and accessed through DAV-API, or else bespoke storage may be used. Any results are written to a temporary content store location, or passed as arguments to the Template Renderer. The latter is called as a final stage, to turn the results into appropriate output text. In this case, templates are used purely to separate business logic from look and feel. Any of the other *Athens* services, for example, publishing, may be called from the application.

The procedure is illustrated in Figure 25. Note that, although the application is here pictured as a monolith at the servlet level, it is obviously good practice to separate out business logic into modules for reuse and possible transaction processing.

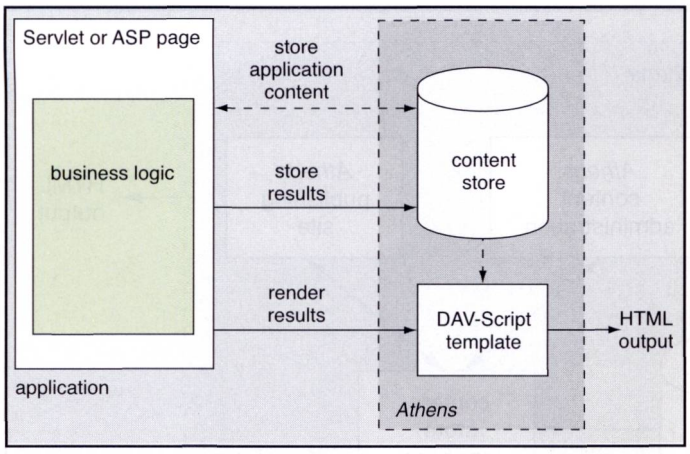


Figure 25: Content Services Approach to Integration

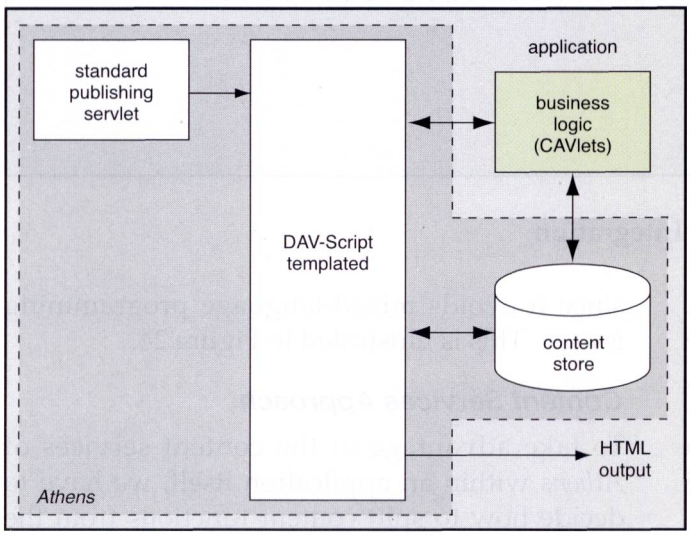


Figure 26: DAVlet Approach to Integration

DAVlet Approach

Alternatively, the application may be viewed primarily as a specialized publishing system, and business logic may be embedded in *Athens* itself. Thus, a page is rendered in the normal way, but the template now also includes elements of bespoke business logic.

DAV-Script itself is not designed for complex programming tasks, so any significant processing will be packaged as DAVlets; i.e., Java (or COM) routines callable by templates. This procedure is illustrated in Figure 26.

Conclusions

In the above we have seen how ICL's Content Management Solution Kit (*Athens*) tackles the many challenges posed by large and/or complex electronic publishing systems, or indeed by any e-Business application with significant on-line content.

Though novel in many respects, *Athens* has effectively been evolving as long as e-Business itself. It draws on proven techniques developed in earlier ICL content-management technologies used in dozens of solutions for major companies and household brandnames. It also benefits directly from ICL's wealth of experience in designing, building and operating such solutions, which cover a wide range of publishing, commerce and other e-Business applications.

However in many ways the *Athens* story is just beginning. In the solution kit philosophy, added-value software components have an important role, but of equal importance is all the accumulated *knowledge*; i.e., experience, know-how, collateral, human contacts, reference sites, worked examples and so on — which has been gathered during use of the solution kit (or its predecessors) in real customer projects. Thus, while *Athens* provides a rich set of tools and building blocks for generating a wide range of content-related solutions, there is plenty of scope for creating "bigger blocks" to encapsulate relevant knowledge and expedite the solution life-cycle to an even greater degree.

Already we are witnessing innovative approaches to real-world applications such as portals, document libraries, Webzines, and discussion groups, based on *Athens* features. Some even involve applications outside what would normally be considered "e-Business". Over the coming months we expect to capture these as design patterns, macros, and pre-canned schema, as well as complete *pro forma* sites capable of being configured and combined to satisfy a range of application requirements, or at least to provide a good foundation on which to build.

This activity will not be limited to the core *Athens* development team. We will encourage submission of useful feedback and case studies from *Athens* users, amassing an ever-growing body of reusable knowledge not normally available to orthodox *product* vendors, and made possible because *our* customers are also our

colleagues, partners, and sponsors. All will be made readily available and approachable through a world-class Content Management and Publishing system!

The ultimate beneficiaries will, of course, be ICL's own customers, investors, and partners.

Biography

Paul Duxbury is currently a Technical Design Authority within ICL's Electronic Business Services division, and is responsible for EBS's Content Management and Publishing architecture. This has included the initial design and prototyping of *Athens*, and more recently its application as embedded technology within other solutions kits. Paul joined ICL in 1973 after graduating in Mathematics from Imperial College. Working first in West Gorton, he soon moved to Kidsgrove where he has since remained. During this time he has been involved, mostly as lead designer, in a wide range of projects and technologies including operating systems development (VME, DRX, CDOS, and UNIX), ICL's DRS20 and DRS300 series with its DRSNET networking, general operating systems security, distributed systems management, Transaction Processing, legacy systems integration, High Availability systems, and most recently Web technologies. He joined ICL Multimedia Solutions (now EBS) in 1997. Paul was appointed an ICL Distinguished Engineer in 1998 and is the holder of a number of patents.

Regulatory Aspects of Electronic Commerce

Tom Cunningham

ICL, Kidsgrove, UK

Abstract

This paper is intended to provide an outline of the current state of some key aspects of legislation relating to electronic business and to suggest some guidelines for solution developers working in this field. It does not set out to provide an exhaustive or definitive account of the existing or emergent regulation in the arena of electronic commerce. This is a fast-moving and, at least for the moment, elusive area of the law which is likely to take some considerable time to stabilise, especially given the complexities caused by the intrinsic facility of electronic commerce systems to operate across international boundaries. The UK Electronic Communications Bill, for instance, has seen many changes en route through its consultative stages, as well as the postponement of some issues which remain to be addressed in future legislation. Also, the legacy of about 1,000 years of existing law already applies in every state in which the European Directive on Distance Selling operates. Reconciliation of the existing statutes with the evolving requirements of electronic trading into a coherent international whole seems a challenging objective, even without considering that electronic commerce extends across recognised boundaries so easily that it amplifies many problems which can usually be overlooked as being of little or no substance in the real world. The challenges posed by the "virtual world" of on-line business have not been comprehensively addressed so far and indeed it may take some years before workable agreements and regulations have been put into force.

It would appear more practical and useful to illustrate some respects in which Internet solutions are likely to be affected by the regulatory aspects of electronic commerce and to offer some suggestions as to what considerations might be applicable when providing solutions in these areas.

Introduction

The expansion of electronic commerce has been attended by much confusion and deliberation over regulation, which has posed many questions to solution providers and governments which are only now in the process of being addressed. Clarification is likely to take some considerable time, as I.T. is evolving too quickly and now has too pervasive an effect on the economy and society for definition of adequate regulatory frameworks to be a straightforward affair. Nonetheless, the continued growth of on-line business and increased consumer awareness of the issues and concerns in this area make increased regulation inevitable.

While the finer points of legislation continue to be discussed, one opportunist viewpoint would suggest that, in such a large, fast-moving and evolutionary environment as electronic commerce, it might be easier and more profitable simply to ignore the various statutes and directives and to run the risk, perceived as slight, of being caught and facing the consequences. Consultancy and development services which set out to provide guidance and best practice on regulatory aspects of electronic business would, by this reasoning, amount to a pointless expense. The question might be asked as to why customers should pay ICL, or anybody else, the cost involved in maintaining awareness of issues and requirements in such an intangible area when providing a particular business solution?

The answer is that in choosing not to pay, customers are courting considerable risk — and professional I.T. partners are well placed to advise on just how substantial this risk is becoming as the regulations evolve. The likely consequences of transgressing regulations can be severely damaging to a business — heavy fines are a

predictable outcome, but by no means the only potential pitfall. Businesses choosing to ignore the regulations could be faced with a mass of civil claims, if individuals sue in response to malpractice. Such failure to observe the regulations could result in contracts being rendered unenforceable, thus undermining business stability, while extreme breaches of the regulations could result in a business being closed down. Risks are not confined to the unscrupulous trader, as a naïve neglect of security considerations can result in susceptibility to hacking attacks which can cripple on-line operations. Along with these risks runs the probability of exposure to the further damage to reputation which comes with the inevitable bad press. Even if a system appears to be distant from the front line of regulatory concerns, the fact remains that if the system plays a part in helping someone to make a living, then it is de facto mission critical and merits appropriate attention and assurance. Customers need protection and assurance which a competent I.T. partner can provide in the form of awareness of the territory, vigilance against common error and assimilation of best practice.

Applicable Legislation

Conducting business over the Internet is a relatively new way of operating for most companies and one which continues to create legal precedents that test, and sometimes fall outside, current trading legislation. Both the UK Government and the European Parliament and Commission are keen to ensure that the growth of on-line business is not hindered by the law, but at the same time they have a responsibility (recognised as a priority) to ensure that consumers' rights and government revenues are not adversely affected by use of this medium for financial transactions. To support this objective, a number of Acts, Directives and Initiatives have been (or are being) produced, some key elements of which are outlined below.

UK Data Protection Act 1998

The 1998 Data Protection Act is based upon the 1984 Act, modified in some areas in recognition of the European Directive's requirements as formulated in response to the expansion of on-line activity over recent years. The Act's intention is to protect the interests of individuals, which it sets out to achieve by the introduction of strict controls on the use of personal data. Any data held by an organisation which can be associated with, or used to identify, a living individual is subject to the Act. The Act also extends the scope of civil liability further than has previously been recognised. Under its terms, data subjects (or individuals) are at liberty to sue data processors directly if they consider their rights to have been violated.

The substance of the Act is well condensed in a set of eight principles, presented below. Square brackets indicate deviation from the exact text of the Act, in the interests of clarity.

1. Personal data shall be processed fairly and lawfully and, in particular, shall not be processed unless the data subject has given his consent to the processing or the processing is necessary for the performance of a contract to which the data subject is a party or at the request of the data subject with a view to entering into a contract

2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.

3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.

4. Personal data shall be accurate and, where necessary, kept up to date.

5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.

6. Personal data shall be processed in accordance with the rights of data subjects under this Act.

7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.

8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the

rights and freedoms of data subjects in relation to the processing of personal data.

These principles have substantial implications for on-line businesses. The requirement for explicit consent to be given for specific uses of individual information, for instance, means that consent mechanisms are essential for commercial web sites seeking to accumulate customer profile information. A basic requirement of the European Directive, which helped to shape the Act, is that unambiguous consent is required from customers to allow their details to be used for specified purposes by the gatherer. This runs counter to the traditional practice which tended towards 'opt-out', or implicit permission, for further use of personal information. The impact on business processes can be widespread. Once obtained, the status of user consent must be tracked and the information protected. Sales and marketing operations need to be reviewed to ensure that provisions of the Act are not breached in sharing access to this data and I.T. and information security processes need to be reviewed when incorporating the provisions. Looking beyond the organisation, contracts outlining the protection of enterprise customer data and penalties for violation should be established with business partners and service providers who might have access to information at this level.

The Act imposes distinct responsibilities upon the data controller and the data processor, and it is essential that all parties are aware of their separate roles in supporting conformance to the principles. I.T. partners' management of their own liability needs to accommodate education of their customers to the specific demands which the legislation places upon them, and identified responsibilities need to be agreed and made explicit in contracts to ensure coverage and avoid ambiguity.

Electronic Communications Bill

The Bill was presented to the House of Commons in late 1999, with Enactment planned for April 2000. It is divided into three main parts:

Part I: Cryptography Service Providers concerns the arrangements for registering providers of cryptography support services, such as electronic signature services and confidentiality services

Part II: Facilitation of Electronic Commerce, Data Storage etc. makes provision for the legal recognition of electronic signatures. It also facilitates the use of electronic communications and the electronic storage of information, as an alternative to traditional means of communication and storage

Part III: Miscellaneous and Supplemental amends and supplements part of the Telecommunications Act 1984. The proposed new provisions are concerned with the modification of telecommunication licences other than in pursuance of a reference to the Competition Commission.

Part II is the most relevant to on-line business solutions. Its objectives include recognition of electronic signatures as legally valid, and facilitation of the use of electronic communications and storage over traditional methods. This latter is achieved by giving ministers the power to modify current legislation to accommodate electronic communications where only paper or other specified means were previously accepted, explicitly redefining the legal interpretation of terms such as 'document' and 'communication' so as to legitimise the use of electronic media. Adoption of the Bill is expected to lead to substantial savings for business in transactions with the Government, owing to the relative speed and economy of conducting business via electronic means.

In its current form, the Bill does not impose any explicit requirements on solution providers, but the changes likely to be introduced through secondary legislation and court decisions, as a consequence of its facilitation measures, suggest that it would be sensible to keep abreast of its progress into statute.

The European Directive on Distance Selling

The European Directive on Distance Selling (Directive 97/7/EC) was originally intended to cover traditional distance selling methods, such as mail order and telephone order, but it has made specific provision to accommodate transactions using new electronic media. The Directive consists of a series of recommendations which need to be implemented via national legislation on the part of each of the EU member states,

embodied in the UK in the Consumer Protection (Contracts Concluded By Means Of Distance Communication) Regulations 2000. Some of the on-line business related aspects of the Directive concern consumer rights and supplier obligations, and are generally protective of consumer interests. Under the terms of the Directive, the consumer has a right to receive 'written confirmation' of order terms and conditions, the 'main characteristics' of the goods ordered, and the identity of the supplier, unless this information has already been supplied on a 'durable medium'. He also has a right to withdraw from commitments 'without penalty and without giving any reason', for a period not exceeding seven working days after making a contract. Suppliers are obliged to fulfil any order within thirty days of receipt of order receipt, unless a specific agreement to the contrary has been made. In addition, the Directive proposes an option for member states to place the burden of proof of compliance on the supplier.

Although the detail of national legislation issuing from the Directive will vary from state to state, the Directive ensures a level of consistency across the EU. This should facilitate common understanding of the considerations of electronic trading which might help remove lingering concerns in the public mind over reliability and assurance.

Disabilities legislation

While no specific legislation relating to disabilities and use of the Internet has been proposed, on-line services fall within the scope of the primary piece of legislation in this area, the Disability Discrimination Act 1995. Part III of the Act details the law relating to service providers and how it is unlawful for a provider of services to discriminate against a disabled person. The Act places a clear responsibility upon service providers to ensure that they are not specifically excluding disabled persons by the way in which their service is being provided.

Exactly how this is likely to be interpreted for the purposes of electronic business solutions needs further definition, which is likely to come out of the current proposal to establish a Disability Rights Commission by way of the Disability Rights Commission Bill which is before Parliament at the time of writing. The Commission, once instituted, will serve as a point of reference for good practice in the treatment of disabled people. One of its objectives is to prepare statutory codes of practice providing practical guidance on how to comply with the law.

Non-legislative initiatives are also providing a lead in the establishment and promotion of best practice in solution design. Among the key contributors are the RNIB Campaign for better Web design and the W3C Web Accessibility Initiative (WAI). The W3C's commitment to lead the Web to its full potential includes the promotion of consideration of usability for people with disabilities. The Web Accessibility Initiative (WAI), in co-ordination with organisations around the world, is pursuing accessibility of the Web through five primary areas of work: technology, guidelines, tools, education & outreach, and research & development.

The Euro

Since 1 January, 1999, the euro has been the common national currency of eleven member states of the European Union. Their own currencies are now known as "national currency units" (NCUs) and are sub-denominations of the euro. These states represent 300 million people, 19.4% of world GDP and 18.6% of world trade.

By 1 January, 2002, all enterprises in the participating countries must be able to conduct all business and to function in every way using the euro. Adoption of the new currency involves two major changes for businesses in participating countries. There will be a transition period during which the euro and NCU will operate side by side, when dual price information will need to be provided for products, followed by a final swap over to a single currency (euro) operation. Over the transition period and for a limited time afterwards, on-line shoppers will need guidance on the value of the euro against their more familiar NCU. Even after full transition has taken place, dual pricing may remain a consideration as there may still be a need to maintain historical data in the original currency units for a predefined audit period, possibly for reference by the tax authorities.

For non-participating countries the euro represents a new and significant foreign currency. However, unlike other foreign currencies its existence and growing use may necessitate the adoption of

dual pricing, just as has been introduced in the participating states. This will be especially the case where it is felt that a significant amount of business may be coming from euro-zone countries where familiarity with the euro will be more commonplace.

Taxation

Taxation is a broad categorisation which covers a variety of secondary issues related to conducting business electronically. Electronic commerce has implications for VAT, other sales and consumption taxes, tariffs, duties, quotas, and possibly profit taxes and personal taxes as well. The anticipated scale of growth of the electronic business economy is likely to accelerate the determination of a set of tax laws specifically for Internet trade but, for the moment, this remains relatively untrodden ground upon which little resolution or harmonisation has been achieved.

In the UK, the Department of Trade and Industry is now consulting on the implementation of the European Distance Selling Directive, which includes a set of proposals for taxing Internet sales, with a view to implementation by the middle of 2000. Implementation of a European solution needs to make provision for regulations applying to non-EU states, so as to make adoption by other countries with a national VAT system (in effect, most developed countries apart from the USA) relatively straightforward. This might provide the EU with a lead in setting the standards for global regulation outside the USA.

Design Implications

Consumer information

Site design needs to recognise the importance of availability of consumer information from the perspectives of usability, customer reassurance and confidence, and conformance to regulations. Solutions should endeavour to provide sufficient information to answer common consumer questions covering issues such as display of pricing, methods of payment, method of delivery, security, contact information, and consumer rights. A considerable amount of information needs to be accessible to the user, so it is important to structure access to this, within the context of the operation of the site and the course of the transaction, to ensure that relevant supporting information is available at appropriate points in the course of the user's visit to, and purchase from, the site — see the above table.

Flow of Events	Information provided
Entry to site	Who you are Access to standard Ts & Cs, Warranty Policy & Statutory Rights Data Protection Policy
Display goods	Price (and currency) information
Purchase request	Payment Methods Security Policy Delivery Information
Request User Details	Data Protection policy
Offer to sell	Returns Form Assurance & Guarantees Complaints Policy Contact Information
Confirm purchase	Receipt confirmation

Terms and conditions

Terms and conditions of the contract should be presented so that the shopper has the opportunity to examine them in detail if desired, before the transaction is concluded. In practice, this need not involve any conspicuous excess of 'red tape' — presentation of a button which can be clicked to access the terms and conditions on the same screen as the 'click to confirm purchase' button may be adequate, although some commentators consider that the customer must actively signal agreement to terms and conditions in order for them to be binding. The concern here is that terms and conditions which cannot be clearly accessed until after the purchase is confirmed may be regarded as null and void, and may even invalidate the contract of sale. The terms and conditions of the site operation (in common with all supporting information) should be presented in plain English and should include the information that these terms do not affect the shopper's statutory rights.

Price

Items should be labelled clearly and multi-currency considerations should be taken into account. On sites where prices are liable to change, it should be made clear to the consumer which price applies to the purchase — for example,

specify that the customer will pay the price displayed at the time of the order, as displayed on the order summary page. Total charges, itemising the cost of the goods plus postage and packing, should be clearly displayed on the order summary page. As a safeguard against disputes, site operators should ensure that records of prices are maintained and are accessible for future reference or for production as evidence.

Payment and security

The standard payment route will conventionally provide a lightweight, rapid access mechanism for submission of the payment details. Supporting information should be accessible on how payments are made, explaining the security mechanisms in operation, how transactions are protected, any guarantees available to the customer and listing the payment methods available (Visa, Master-Card, American Express etc.). A brief explanation of the storage and use of consumer information submitted to the site should be provided, incorporating a statement on adherence to Data Protection legislation.

Delivery

On purchase, information should be presented on how delivery will be made and within what time scales. Appropriate caveats should also be presented to avoid creating the impression that delivery is guaranteed.

Contact information

Information should be made available advising shoppers about contacts for further information on issues such as order tracking, order cancellation etc. All addresses (email and "snail mail"), telephone numbers, customer help line information and so on should be presented to the shopper. Corporate registration details (VAT registration and Company registration numbers) should also be available.

Guarantees and assurance

Policy on guarantees, guidelines and processes for handling any instances of consumer dissatisfaction should be provided as supporting information directly accessible from the confirmation of purchase screen and as background information from elsewhere on the site. Guarantees, if offered, should be clearly presented and their scope made plain. Any refund policy should be stated along with the conditions for return of the purchased goods. The procedure for shoppers to alert the solution owner to any complaints or issues should be described along with the site policy and process for the resolution of issues.

Consumer law

Commercial site operators should be aware of the consumer legislation in effect in the country of residence of the customer and should be involved in the design specification process so as to ensure that the solution operates to their requirements with respect to the legislation. Supporting information need not go into detail but should provide an outline assurance that the site conforms to the appropriate requirements. The underlying activity required on the part of the site operator to confirm assurance is not a trivial task, as is illustrated by the breadth of UK legislation which includes the Sale of Goods Act 1979, the Supply of Goods and Services Act 1982, the Consumer Credit Act 1974, the Trade Descriptions Act 1968, the Unfair Contract Terms Act 1977 (supplemented by the 1994 Regulations) and the Consumer Protection Act 1987.

Security of Information

The Data Protection Act places an unequivocal emphasis on the need for information security and protection of the confidentiality of the information held. All electronic business solutions need to adhere to the principles embodied in the Act and to prove their conformance by appropriate means (such as audit trails and formal published procedures). Careful thought needs to be applied to the questions of what information can reasonably be kept for the required purposes and how long that information needs to be kept. These cannot be arbitrary decisions and need to be related to meaningful and defined requirements such as refund periods, return of goods, guarantees (proof of purchase), or financial audit purposes.

Protection of individual rights is a key objective of the Act, and its provisions allow citizens the right to request access to any data held in connection with themselves, and also the right to request modification of that data if there are any issues over its accuracy or relevance to identified requirements. Time constraints are imposed on the processing of any such requests, so organisations need to be equipped to be responsive to any issues raised in respect of data held.

Specific consideration needs to be given to the collection and handling of customer information based on personal data, information about shopper purchase history, or information about a shopper's navigational history around the site. The Act stipulates that the Data Protection Commissioner should be informed of any "new categories of data subject", or "collecting data in categories other than those already held". This may be seen to run counter to one of the underlying objectives of the analysis of customer data, which is to develop new views and interpretations of available information which may provide further business opportunities. In particular, the introduction of supplementary data based on the analysis of customer history which might be intended for internal use, such as a recorded observation that an individual may represent a credit risk or may not be trustworthy, needs to be regarded with extreme caution.

Accessibility

The accommodation of visual disability is a key problem for Internet solution and service providers, which in some cases is already being solved by solution designers applying the appropriate design principles. The BBC system, for instance, has the facility to convert a Web page into simple large font text which can be read more easily by the visually impaired. Other forms of disability, which might be borne in mind, include physical handicaps. For example, systems which are heavily reliant on keyboard input may pose difficulties to the severely arthritic.

A set of accessibility guidelines have been produced by the Web Accessibility Initiative (WAI), which include the following:

- provide text equivalents for all non-text elements (such as images, animations, audio, video)
- provide summaries of graphs and charts
- ensure that all information conveyed using colour is also available without colour
- organise content logically and clearly
- provide alternative content if using features (e.g., applets or plug-ins) which may not be supported.

International and Cultural Considerations

The association of accessibility concerns with the requirements of disability discrimination legislation is well recognised, but electronic business solution designs also need to recognise other forms of legislation against discriminatory practices, along with ethical and social conventions, which exist in various forms in various countries. Age discrimination, for instance, is an issue in Scandinavian countries where direct advertising to minors is forbidden. Anecdotal instances of social peculiarities abound — perfumes cannot be sold in Saudi Arabia (because of their alcohol content), and sexually explicit advertising (even as moderate as the depiction of models in swimsuits) is notoriously poorly received in Islamic nations. In Germany, "special offers" are forbidden since they are perceived as detrimental to the interests of other vendors trying to obtain a fair market price. The general point to be appreciated is that there are few safe assumptions where international culture is concerned, and many conventions which are regarded as innocuous to the Western observer can carry the risk of causing offence or even violating the law in some corner of the globe.

To adopt a "lowest common denominator" approach and set out to avoid any possibility of offence could easily result in suffocation of imaginative design. A more practical approach would be to assess the target market, identify any potential issues and stipulate appropriate exclusions and disclaimers prominently on the site. If it is made clear that a site is only intended for operation in specified countries, and if this caveat is reinforced by a mechanism which will ensure rejection of orders originating from the excluded regions, then the system can be demonstrated as operating with reasonable controls and can be defended plausibly against charges of mischief or wrongdoing. Assurance against such charges is principally the responsibility of the site operator, but the solution provider might provide guidance in highlighting any potential for misinterpretation or offence.

Currency

Conversion calculations between NCUs and the euro, as well as between non-participating countries' currencies and the euro, will be

required to support dual currency representation in electronic commerce solutions. Inaccuracies inevitably occur due to rounding when values are converted from one currency to another, then back again, with the inaccuracy being compounded if this two-way conversion is iterated in the course of processing a transaction. Strict rules need to be applied to currency conversion algorithms and advice has been provided by the Euro Working Group, an independent I.T.-based organisation, on methods and tools for use in transition and conversion.

One key design objective should be to ensure consistency of prices displayed at the point of purchase, on confirmation and on receipt, to avoid consumer confusion and poor perception of the site. It is also important to ensure that where prices are displayed in dual currencies, there is no confusion over which price relates to which currency. More materially, there is a pressing business reason to eliminate scope for confusion as, during the transition of Euro-zone countries, electronic business solution operators may want to accept payment in either currency, thereby introducing the potential for loss of revenue due to conversion errors.

Taxation

The complexity and lack of harmonisation of international tax regulation makes this a difficult area in which to offer any substantial guidance to solution designers. While electronic commerce solution providers need to be equipped to calculate and collect taxes applicable to the countries in which their solutions operate (where the products sold are used or consumed), it is the commercial operator's responsibility to define the country of residence and to supply a definition of the taxation rules for implementation within the solution. Separate consideration needs to be given as to how taxation is applied to goods sold to shoppers in different countries, operating under different tax regimes. Taxation laws relating to sites operating in countries outside the European Union are likely to require case by case investigation with the support and guidance of legal specialists.

The solution provider can best assist in two ways, by ensuring that supporting information on applicable tax considerations is available on the site and by providing comprehensive backup and audit facilities to assist in any future requirement for the analysis of transactions. In order to guide and inform the consumer, an electronic commerce site should present information on how taxes are applied to purchases made on the site, as well as clearly presenting any liability or disclaimer statements. The tax implications of on-line transactions need to be presented clearly, as individual consumers are highly unlikely to be familiar with the complexities which might be involved. Rules will vary from country to country, and consumers may not be aware of the scope for variation which results from the international nature of Internet trading.

Also, as the Internet tax laws develop, and as the volume of on-line trade continues to grow, a need for organisations and government revenue bodies to have access to comprehensive audit records is likely to become standard. Forward-looking system designers should recognise this and should accommodate the requirement in solution design and in operational procedures.

The bulk of the responsibility for defining the tax rules governing the operation of a site must rest with the site owner. A simple conclusion to draw and communicate is that taxation is a dangerous area in which to take chances, making expert consultancy an essential part of adequate preparation on the part of the commercial operator.

Liability

The advent and evolution of regulation in the arena of the Internet and electronic business, regardless of the specifics of existing or emergent law, will make it far more likely that I.T. companies become involved in issues with legal ramifications, with the obvious possibility of being drawn directly into legal disputes and proceedings. While civil claims have always been a potential outcome in worst case disagreements with clients, until recently it has appeared almost impossible that an I.T. organisation would be involved in criminal proceedings as a result of attempting to conduct routine business. Offences involving the use of computer systems in the perpetration of fraud, distribution of pornography, or criminal damage (the legal view of "hacking") are no longer the curiosities which they were only a few years ago.

The Data Protection Act introduces further criminal offences for which companies may be held responsible, and the terms of the Act suggest a requirement for a similar level of attention and diligence to that which is routinely conducted, for example, by organisations in respect of the Health and Safety at Work Act.

This extension to the scope of criminal liability is likely to demand provision for the availability of a different level altogether of supporting evidence from that which might have been required to mount a defence against a civil lawsuit for compensation relating to malfunction in a software product. Information relating to all aspects of system operation might be regarded as relevant to proof of system effectiveness. The more auditable the system, the less problematic will be the collection of evidence needed to support any case. Maintenance of records or logs of every single access to a system is likely to become a standard requirement. This can even provide unexpected benefits, as was proved in a recent case in which ICL was asked to assist the police by providing access to server logs to enable tracking of an individual suspected of trafficking in child pornography on the Internet. With support from ICL technicians, the suspect was traced, and was later charged and brought to justice.

The routine and automated collection of system log data is more reliable than any ad-hoc mechanism and as such should be preferred. Backup procedures need to be comprehensive and should be tested thoroughly. The network and host system configuration should be monitored using tools to check for any alterations, to guard against malicious interference. Although this may seem an excessive and expensive approach, it should be weighed against the potential cost of collateral damage to the business involved in defending criminal actions. Apart from the adverse publicity which would attend involvement in a criminal trial, and the immediate cost of professional legal representation, the practical consequences of losing the services of technical experts assigned to act as expert witnesses can have a significant adverse effect on productivity. The burden of proof is higher in criminal proceedings than in civil proceedings, and cases involving any level of technical complexity can be expected to run for weeks rather than days.

Taking Care

Many of the suggestions made in this overview amount to a simple application of common sense in the absence of any precise direction as regulations governing on-line business take shape. Where no specific and authoritative advice is available, making the possibility of error almost unavoidable, then generalised advice to err on the side of caution may be worthwhile.

A useful analogy may be drawn with the laws relating to motor accidents and the care of motor vehicles. Nobody is sent to prison for having an accident; rather, they are punished for being judged as having contributed to the cause of an accident. The emphasis is on the contribution to the cause, as is illustrated by the fact that one can be held to account for driving a vehicle in a condition which is considered likely to increase the chances of an accident occurring, even if no accident actually takes place (the consideration of condition applies to the driver as well as to the vehicle). So there is a clear, two-fold advantage to awareness and the appropriate precautions; if one understands the legal requirements for vehicle maintenance, and if one looks after oneself and one's car, one is less likely to be involved in an accident in the first place. If one is unfortunate enough to be involved in any incident, one's defence will be the stronger if one can be shown to have taken account of the best practice in care and control of oneself and one's vehicle.

To return our perspective to the sphere of I.T. development, this endorsement of awareness and the appropriate precautions might be translated into a recommendation to reintroduce some familiar engineering processes which have fallen (at least to some extent) into disuse in response to demands for more rapid and lightweight development and delivery schedules which have accompanied the "Internet age". The proliferation of personal Web sites, hobbyist publications and amateur Web designers have helped to shape an illusion that Web site design is a straightforward affair, coupled with an assumption that even the most complex sites can be produced by a competent team in a matter of a few weeks. Techniques, such as prototyping, are heavily used because of their ability to demonstrate results quickly, a particular advantage where visual and design aspects are at the forefront of the customer's mind. However, this demonstrable

progress can easily mask underlying flaws which are only likely to be exposed by more formal assurance and testing techniques. Reuse of already developed technologies or components is strongly promoted as a means of reducing the cost of development, sometimes at the expense of adequate testing of integration into specific environments. Experience of many projects would illustrate that questions which have been taken as "no-brainers" have turned out to be anything but. Detailed modification control procedures, access and action journals, test specifications, test plans and test result documents may seem like products of a bygone age, but they only ever existed to provide a level of assurance which is often signally lacking on Internet projects.

Another difficulty of Internet development is that the World Wide Web is a mass market, not an educated environment sensitive to the complexities of highly technical bespoke solutions. Internet products are released into the hands of an unknown, potentially untutored, and sometimes even hostile audience (cases of mischievous or malicious tampering are commonplace). The traditional practice of releasing software product along with a user's guide is inapplicable, as nobody knows who the users are, and their numbers may run into millions. Unforeseen aberrations in software operation can no longer be dismissed by reference to user documentation — the traditional support dictum 'RTFM' is redundant, since with Internet systems there is no manual, and with Internet users there are no rules.

Processes such as change control, version control, production of test plans and test logs, regression test and code inspection have all been compromised by the imperative for quick delivery, and in some less substantial developments have even been overlooked completely. These aspects of the development process need to be reviewed and refined with the particular requirements of Internet development and the obligations imposed by electronic commerce legislation in mind.

Conclusions

Although definitive and comprehensive regulations governing electronic commerce appear to be some way off, it remains the case that with a little forethought and with an appropriate awareness of the issues outlined above, solution providers can make some provision for the effect of emergent legislation on on-line business operations. What seems clear is that ignorance of the issues will lead to difficulties and complications, if not to straightforward violation of the law. In an area where there are few absolutes, and where custom and practice has yet to be established, a good deal of weight is likely to be attached to the degree of effort and good intention shown by the commercial operator. A business which can be shown to have made an effort and which has achieved what is generally felt to be a level of good practice might avoid heavy punishment, or might be exonerated, even if it can be proved that it has deviated from the letter of the law. On the other hand, if a business displays a conspicuous and wilful ignorance of the regulations and considerations which apply to its area of operation, it is likely to find itself facing a substantial and punitive response from the courts. Also, on-line commercial fraud has grown alongside on-line commerce, and there is evidence that despite continued assurances, customers continue to harbour suspicions about the security of doing business on-line. An on-line business not only needs to be trustworthy, but also needs to be seen to be trustworthy, and successful self-regulation is an important element of building trust.

Acknowledgements

Specific thanks are due to Dick Emery, for his illuminating views and his guidance through the intricacies of legislation; to Andrew Hobson, for research into the substance and implications of European initiatives; to Jonathan Smith, for direction on legislative detail; and to Steve Picken, for advice and encouragement.

Biography

Tom Cunningham joined ICL in 1988, and has occupied various technical and operational roles for both the ICL and Fujitsu organisations. His involvement in electronic business dates back to 1997, when he joined the fledgling Internet Applications group, later to become Electronic Business Services, as a technical manager. Tom holds an honours degree in Philosophy from Liverpool University and an MSc in Computing Science from the University of North Staffordshire, along with a number of professional accreditations.

Scalable e-Commerce Solutions

Stuart Forbes

ICL, Manchester, UK

Abstract

As demand for e-Commerce increases, businesses need scalable, high-performance Internet solutions, capable of handling heavy transaction volumes and large numbers of users. Many demonstrations have been run which show a very high performance, but these are usually not based on real world solutions. ICL and Microsoft have conducted benchmark tests in Manchester and Redmond to prove the scalability of ICL's interactive shopping solutions based on Microsoft Back-Office applications.

An Internet shopping mall was created and a shopper load simulated. The following results were achieved:

- The number of shoppers that could be supported on a Windows 2000 Web server was 50% more than on a Windows NT4 Web server
- The shopping mall supported 5,400 concurrent shoppers while maintaining a mean response time of less than 1 second. Moreover, test results indicate that the shopping mall will scale linearly up to 10,000 concurrent shoppers
- Under the same heavy load conditions simulated in the benchmark testing, 5,400 concurrent shoppers could be expected to perform just under 7.5 million transactions per day, while 10,000 shoppers would perform more than 14 million transactions per day
- Based on the assumptions used in running the benchmark tests and some estimates of how often shoppers will frequent the mall, these performance results indicate that ICL's Internet shopping solution could support a shopping population of at least 2.7 million people and as many as 5 million people. This is a far higher performance than most shops will ever require.

Introduction

ICL provides managed Internet shopping solutions based upon ICL's Interactive Shopping Solution Kit which uses Microsoft's Back-Office applications — in particular IIS, Site Server Commerce Edition and SQL Server — running on Windows NT4 Server and Windows 2000 Advanced Server.

This paper describes a joint benchmarking exercise between ICL and Microsoft to investigate the scalability of ICL's Internet shopping solutions. The aims of the exercise were:

- to investigate the scalability, limits and robustness of ICL's shopping solution
- to provide sizing information to allow cost-effective sizing of large shopping sites — thus reducing the cost of ICL's solutions whilst maintaining confidence that the shopping workload can be handled
- to offer a real-world workload for Microsoft to demonstrate the scalability of their products on both NT4 and Windows 2000.

The NT4 benchmarking was carried out in ICL's enterprise laboratories in Manchester, UK and the Windows 2000 benchmarking was carried out initially in Microsoft's scalability laboratories in Redmond, USA and then in Manchester.

The benchmark configurations used ICL's enterprise-class Trimetra P2000 servers (4 and 8 way Pentium III Xeon 550 MHz).

As the benchmark results discussed later in the paper show, ICL's Internet shopping solutions are capable of delivering more power and scalability than most shops will ever require.

Interactive Shopping Solution Kit

The Interactive Shopping Solution Kit (ISSK) was used to develop ICL's Internet shopping solutions. It provides a set of components that can be used by ICL to generate genuinely bespoke customer Web sites at reduced cost and time to market. A site can be built using the solution kit that hosts a single shop, a mall or multiple malls. The kit allows the content of the site to be managed by the customer whilst the site itself is managed by ICL. It is based on experience gained from deploying sites for many customers and has been used for a number of important customers such as the BBC Shop.

ISSK is based on IIS, Site Server Commerce Edition and SQL Server. It uses HTML Web Templates to provide the look and feel of the site and ASP pages to provide the business logic. It uses SQL Server to hold the product catalogue, the structure of the site, promotions, shopping baskets plus orders and receipts. SQL Server is accessed using stored database procedures to improve performance. There is also a small amount of use of COM objects and MTS, for example for merging the templates and for processing orders. Site Server is used for search, personalisation, analysis, advertisements and commerce.

The basic structure of the solution is given below in Figure 1, but a fuller description is given in [Picken, 1999].

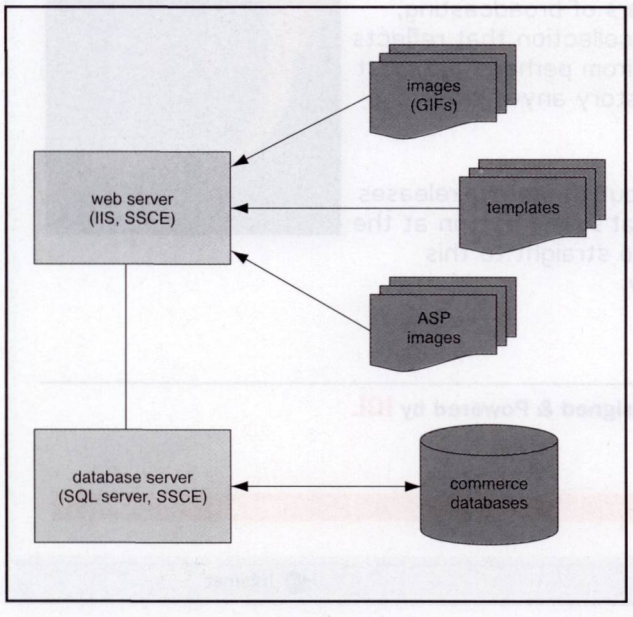


Figure 1: ISSK Structure

ISSK also includes site administration and staging of content by using a separate administration server. The benchmarking exercise was focused on shopping — which provides the bulk of the workload.

Initially the benchmarking was based on ISSK version 1. However, by the end of the exercise ISSK 2 was available and all the results in this paper are based on this new version. ISSK 2 includes many performance improvements, some of which were identified by the benchmarking team, and some of which were planned enhancements. As a result, ISSK 2 supported more than twice the number of shoppers as ISSK 1 with faster response times.

Scaling and Scalability

A fully scalable solution is one where an increase in the hardware resource available results in a corresponding increase in the ability of the solution to do useful work. It is rarely possible to achieve this ideal in practice and a more pragmatic goal is that a scalable solution is one where an increase in the hardware resource available results in a predictable and cost-effective increase in the ability of the solution to do useful work.

The basic hardware configuration for an ISSK solution is given in Figure 2 below.

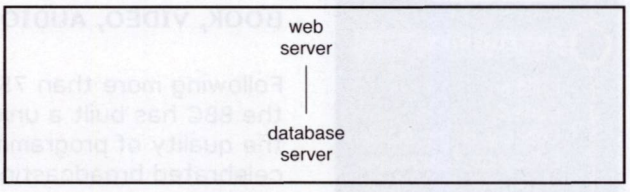


Figure 2: Basic Configuration

It is possible to scale the hardware in two ways:

Scaling up: increasing the power of each server
Scaling out: by increasing the number of servers

Both options were explored in the benchmarking exercise. The system was scaled up by increasing the number of CPUs in each web server from 1 to 4. The system was scaled out by increasing the number of web servers from 1 to 3 and using WLBS (Windows NT Load Balancing Service) to balance the load across the servers — WLBS can support up to 32 servers in a cluster. This is

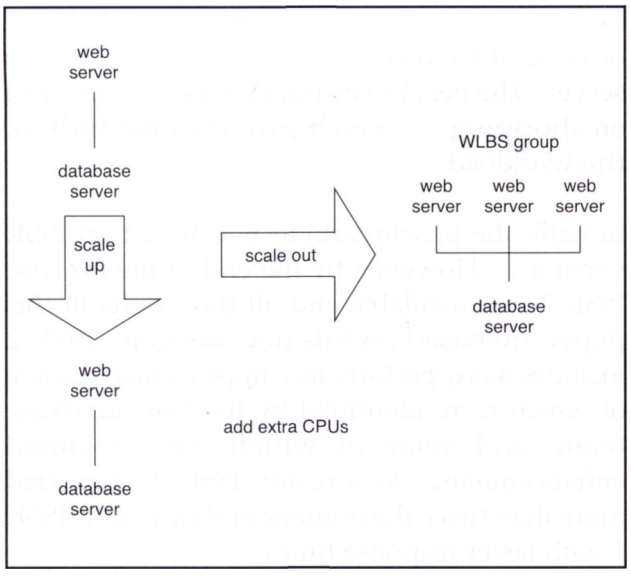


Figure 3: Scaling Up and Out

shown in Figure 3. Further scaling could have been achieved by having more than one database server and by partitioning the shopping mall across multiple servers. Note: WLBS is renamed NLBS (Network Load Balancing Service) in Windows 2000.

During the benchmarking activity many benchmark runs were performed on each configuration with the number of simulated concurrent shoppers accessing the shopping mall increased each time. The peak number of concurrent shoppers, which the system could handle with acceptable response time, was recorded.

However, it is important that an Internet shopping solution can cope with peak demands

Figure 4: BBC Shop

beyond which it has been sized for. So the benchmarking activity also examined the behaviour of the system with numbers of shoppers far higher than that for which the system could provide good response times for all shoppers, to check that the solution remained robust and that the throughput did not drop.

BBC Shop

The benchmark is based on the BBC Shop, a successful deployment of the Interactive Shopping Solution Kit (www.bbcshop.com). Figure 4 shows a screenprint of the home page.

Benchmark

The benchmark modelled a shopping site that consisted of one shopping mall with two shops. These shops were based on the BBC Shop site described above.

The benchmark supports a number of different shopper profiles derived from an analysis of real Internet shops as shown in Table 1.

Emulation Environment

The hardware and software used is described in Tables 2 and 3 respectively.

More information on the ICL Trimetra P2000 servers may be obtained from the ICL website and, in particular, at http://www.icl.com/itservices/servers/servers.htm.

The benchmark environment used in the trials is outlined in Figure 5.

Shopper Profile	% of Shoppers	Transactions	Description
Browser Type 1	20%	18	Shopper browses the site looking at editorial pages, product summaries and product details, but does not perform any searches or add any items to the basket.
Browser Type 2	20%	12	Similar to browser type 1, but looks at fewer pages.
Adder	20%	24	Shopper browses the site and adds an item to the basket, but does not purchase it.
Purchaser	20%	32	Shopper browses the site, adds an item to the basket and purchases it.
Searcher	20%	14	Shopper performs a search and looks at the details of products returned by the search. No item is added to the basket.
Mean		**20**	

Table 1: Shopper Profiles Benchmark

Hardware		
Server	**No.**	**Description**
Web Server	3	ICL Trimetra P2000 PIII 550MHz , 1MB cache, 1GB RAM [1 to 4 CPUs]
Database Server	1	ICL Trimetra P2000 PIII 550MHz , 1MB cache, 2GB RAM, 8 CPUs
Load Generator	2	FJ ErgoPro PII 266MHz 64MB RAM
Monitoring Station	1	FJ ErgoPro PII 266MHz 192MB RAM
Network	1	100Mb Switch

Table 2: Benchmark Hardware

Software	NT4 Testing	Windows 2000 Testing
Web Server	ICL Interactive Shopping Solution Kit 2.0	ICL Interactive Shopping Solution Kit 2.0
	Microsoft Windows NT4 Server SP5	Microsoft Windows 2000 Advanced Server Build 2195
	Microsoft Site Server Commerce Edition 3.0 SP2	Microsoft Site Server Commerce Edition 3.0 SP3
	Microsoft Data Access Components 2.1 SP2	Microsoft Data Access Components 2.1 SP2
	Microsoft Internet Information Server 4.0	Microsoft Internet Information Server 5.0
	Microsoft Transaction Server	Microsoft Transaction Server
Database Server	ICL Interactive Shopping Solution Kit 2.0	ICL Interactive Shopping Solution Kit 2.0
	Microsoft Windows NT4 Server SP5	Microsoft Windows 2000 Advanced Server Build 2195
	Microsoft Data Access Components 2.1 SP2	Microsoft Data Access Components 2.1 SP2
	Microsoft SQL Server 7 SP1	Microsoft SQL Server 7 SP1
Load Generator	InetMonitor 3.0	

Table 3: Benchmark Software

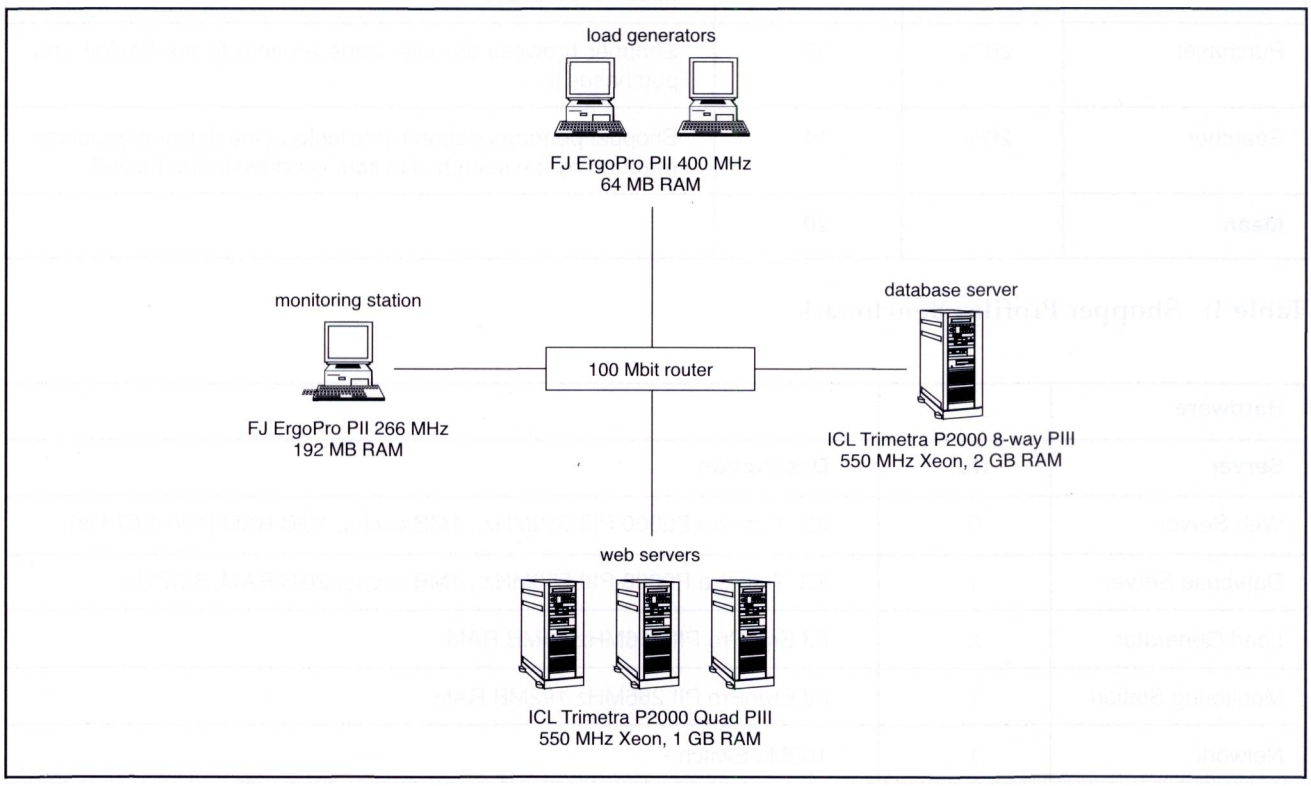


Figure 5: Benchmark Environment

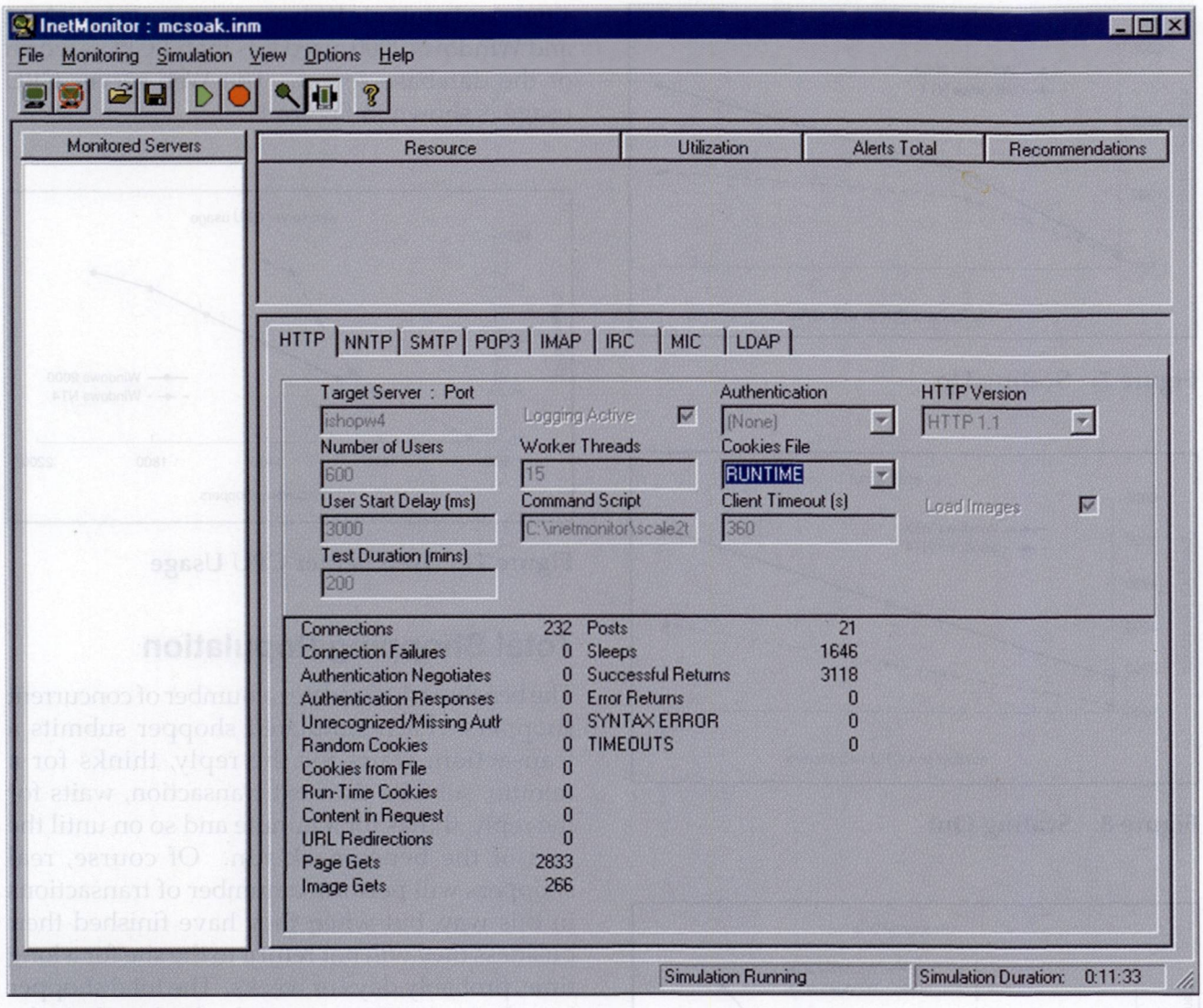


Figure 6: InetMonitor

Load Generation

The load generator used for the benchmark was Microsoft Inetmonitor. This ran on a number of PCs. It was chosen for its ability to reflect accurately the shopper workload, its ease of use and the large number of shoppers that could be simulated on each PC. Figure 6 shows a screenprint of Inetmonitor.

Key Results

The maximum number of concurrent shoppers which could be supported by each configuration while delivering appropriate responsiveness was measured. The criteria used were that less than 80% of the total system CPU should be utilised on each server and that the mean transaction response time should be less than 1 second.

Figures 7 and 8 show how the number of concurrent users that could be supported increases as the system is scaled up and out — for both Windows NT and Windows 2000.

Detailed Performance Information

In the following figures, a more detailed view of the performance of the system is shown while running the benchmark tests. The graphs compare the Windows 2000 and NT4 on the single 4-way Web server configuration.

The Response Time and Throughput as the number of concurrent shoppers increases is shown in Figure 9 and Figure 10.

The CPU usage of the Web server and the database servers was also measured. All the

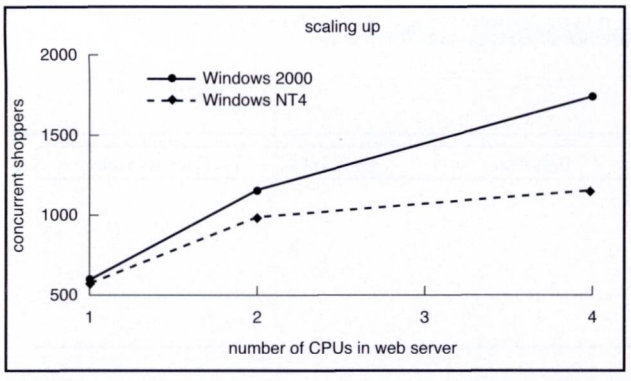


Figure 7: Scaling Up

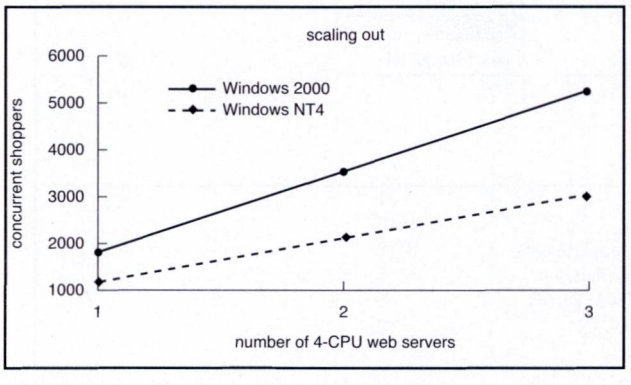


Figure 8: Scaling Out

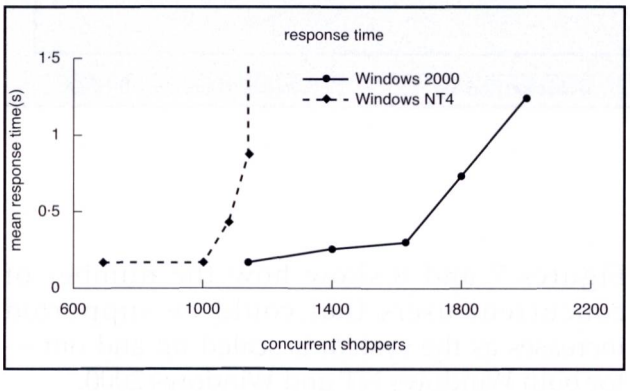


Figure 9: Response Time

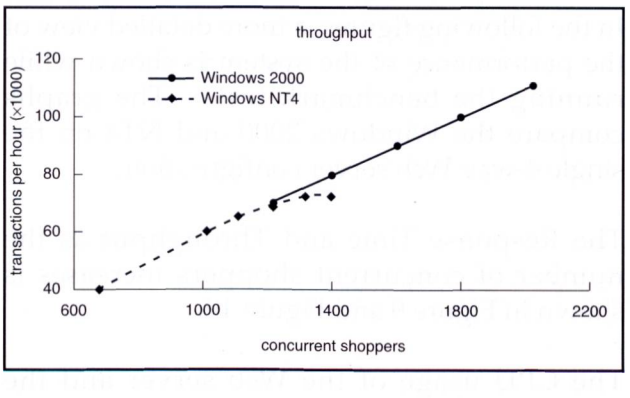


Figure 10: Throughput

single and multiple Web server runs, on both NT4 and Windows 2000, used less than 1 CPU's worth of the database server. The Web server CPU usage is shown in Figure 11.

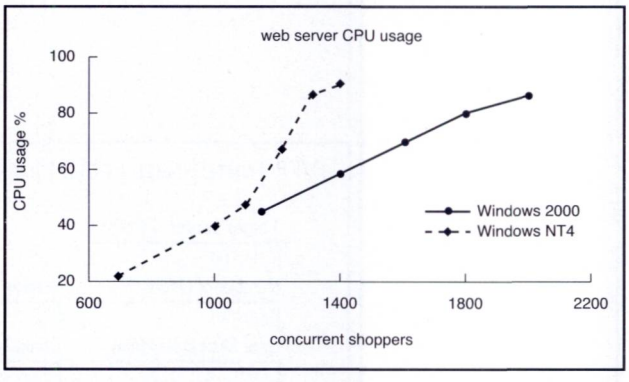


Figure 11: Web Server CPU Usage

Total Shopping Population

The benchmark simulates a number of concurrent shoppers. Each simulated shopper submits a transaction, waits for the reply, thinks for a minute, submits the next transaction, waits for the reply, thinks for a minute and so on until the end of the benchmark run. Of course, real shoppers will perform a number of transactions in this way, but when they have finished their business they will not return to the site for a long time, probably days or weeks. The total shopper population using a site is far higher than the number of shoppers concurrently using the site at any one time.

It is difficult to estimate the total shopper population a site can support, but the following gives a good approximation, and has been supported by analysis of live shops.

The following assumptions were made:

- An average session for a shopper consists of 20 page requests and lasts for about 20 minutes
- Regular shoppers visit the site on average once every month
- The peak 20 minutes of the week handles four times as many shoppers as the average 20 minutes.

This implies that 1 concurrent shopper is equivalent to about 500 regular shoppers.

The argument for this is as follows:

The number of 20 minute periods in one month is roughly 2000

Each regular shopper makes one visit in one month, so that the number of visits per shopper in an average 20 minutes is 1/2000

The peak 20 minutes handles four times as many shoppers, which is 1/500

Because an average visit lasts 20 minutes it equates with a concurrency of 1/500 shoppers; i.e. each concurrent shopper equates with 500 regular shoppers.

Capability of ICL's Solution — 5 Million Shoppers

The benchmark scaled up the number of web servers to three 4-way servers attached to a single 8-way database server. This used less than 10% of the capacity of the database server and there is no reason not to believe that at least six web servers could be attached to a single database server without any significant increase in response time. This allows a web site supporting a single shopping mall to be scaled up to very high shopper populations — see Figure 12.

The capabilities demonstrated by the benchmarking exercise are given below:

- The shopping mall supported 5,400 concurrent shoppers using 3 web servers while maintaining a mean response time of less than 1 second. Moreover, test results indicate that the shopping mall will scale up to more than 10,000 concurrent shoppers by increasing the number of web servers to 6

- Under the same heavy load conditions simulated in the benchmark testing, 5,400 concurrent shoppers could be expected to perform about 7.5 million transactions a day, while 10,000 shoppers would perform about 14 million transactions per day

- Assuming that each concurrent shopper equates to 500 shoppers, then ICL's Internet shopping solution could support a shopping population of at least 2.7 million people and as many as 5 million people. This is far more performance than most shops will ever require.

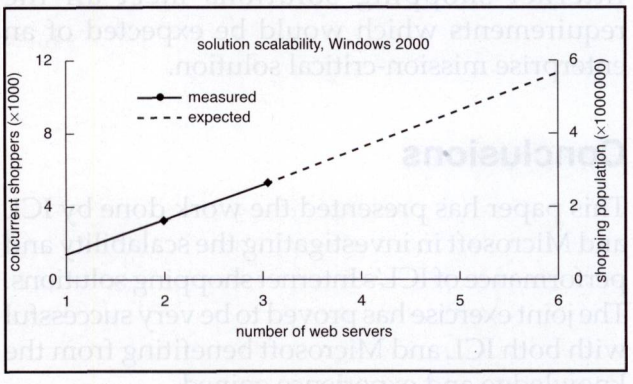


Figure 12: Scalability of Shopping Mall (Windows 2000)

Microsoft Windows 2000

ISSK was very quickly and easily upgraded from Microsoft Windows NT4 to Microsoft Windows 2000. The Internet shopping benchmark was then used by Microsoft in Redmond to validate performance improvements in Windows 2000. Measurements taken on Windows 2000 Advanced Server Build 2195 showed significant improvements over Windows NT4. Over 50% more shoppers can be supported per web server on Windows 2000 than on Windows NT4.

Enterprise Shopping Solutions

The benchmarking exercise focused on the scalability of the solution from the point of view of supporting more concurrent shoppers. It is also important that the administrative and systems management aspects of the solution scale — for example, the processes for back-up and for updating the content of the site must be able to handle large sites and large numbers of shoppers.

In order to produce a full enterprise solution it is important to consider all the other enterprise qualities such as:

- Manageability
- Availability
- Reliability
- Security.

ICL has a long and successful history in the enterprise server business, and is exploiting its knowledge of all these issues to ensure that its Internet shopping solutions meet all the requirements which would be expected of an enterprise mission-critical solution.

Conclusions

This paper has presented the work done by ICL and Microsoft in investigating the scalability and performance of ICL's Internet shopping solutions. The joint exercise has proved to be very successful with both ICL and Microsoft benefiting from the knowledge and experience gained.

ICL's Internet shopping solutions and the Microsoft products on which they are based have been shown to be responsive and highly scalable on ICL's NT servers. It is anticipated that ICL could reproduce these results on other comparable NT servers, but this has not yet been tested.

Internet shopping is still in its infancy and an explosive growth is likely to occur over the next few years. ICL's solutions are capable of supporting more shoppers than most Internet shops will ever need.

Acknowledgements

The author would like to thank all the people involved in the benchmarking exercise: Stewart Sutcliffe, Gill Newbold, Andy Kirk, Mark Coleman, John Ruscoe, Stephen Hoyle, Dave Williams, Steve Picken and Colin Rutland all from ICL and Simon Davies, Robert Barnes, Sally Martin and Carl Carter-Schwendler from Microsoft.

Glossary

ASP	Active Server Pages.
COM	Component Object Model.
ISSK	Interactive Shopping Solution Kit.
HTML	Hypertext Mark-up Language.
IIS	Internet Information Server.
MTS	Microsoft Transaction Server.
NLBS	Network Load Balancing Service.
SQL	Structured Query Language.
SQL Server	Microsoft relational database.
WLBS	Windows NT Load Balancing Service.

Bibliography

PICKEN, S., "Internet Shopping Services", ICL Systems Journal, Volume 14, Issue 1, Autumn, 1999.

Biography

Stuart Forbes is a Systems Architect based in Manchester. He has worked on a variety of development and customer projects specialising in performance. He is currently involved in the design of enterprise Windows NT and Windows 2000 solutions.

Stuart joined ICL in 1986; he is a Member of the IEE and an ICL Distinguished Engineer.

Email: stuart.forbes@icl.com

Public Key Cryptography, e-Commerce's Magic Security Bullet?

Tom Parker

ICL Fellow Emeritus

Abstract

Public Key Cryptography is hugely important. Without it, there would be little serious electronic commerce, there would be no ubiquitous secure global email and, perhaps most importantly for the future, we would be stuck with the fact that if a document is to have legal force, it must be on paper. The basic Public Key concept is simple, but it has given rise to a vast and complex supporting Public Key Infrastructure, or PKI, which depends for its success, not only on getting the technology right, but also on understanding how the needs of the real business world are being represented by it. As we begin to appreciate the difficulties, the original golden boy is beginning to show some tarnish! This paper first introduces the basic technology, then jumps into the deep end of PKI to examine the business and legal issues surrounding its use.

Introduction

Public Key Cryptography (PK) has come of age. It is becoming the most important single security technology in use today.

When PK was first introduced, it was thought that its security properties were simple and that its application would be relatively straightforward. Since then, a huge and complex infrastructure (known as Public Key Infrastructure or PKI) has developed as the subtleties of the security properties and requirements of PK have come to be better understood. In this paper we describe this huge infrastructure and reveal some of the worms in the PKI woodwork, particularly those which arise in the context of e-commerce. We focus on digital signatures, because they are the most problematic use of the technology, and we look at their huge social and technological significance. First though, we introduce the arcane magic of Public Key Cryptography, contrasting it with conventional symmetric cryptography, by explaining how the two work.

Public Key Cryptography and Public Key Certificates

Traditionally, the security of a cryptographic exchange has been based on each party sharing the same secret key value. The sender encrypts and the receiver decrypts using this one value. Anyone listening to the exchange cannot decipher its contents unless they know the key. The technology underpinning this is known as symmetric key cryptography, and it is as old as time.[1]

In Public Key Cryptography, keys come in matched pairs. The sender encrypts with one key and the receiver decrypts with the other. A key cannot be used to decrypt data which it has just encrypted. Knowing one of the keys of the pair does not help in finding the value of the other key. These simple, but quite amazing, properties open a new world of opportunities the importance of which is hard to overestimate. Public Key Cryptography is a computer age invention.

One of the keys of the pair is kept private to a single individual, and is unsurprisingly known as a **Private Key**.[2] The other key, the **Public Key** can be broadcast to the world.

---

[1] It goes back at least as far as the ancient Egyptians, who would wrap a long papyrus strip round a wooden stick, and write on the resulting surface. When unwrapped, the marks on the papyrus were indecipherable. Only someone with a stick of the same diameter (the diameter is the "key") could re-wrap the papyrus, line up the edges as in the original, and decipher the writing.

[2] Sometimes wrongly referred to as a secret key; secrets can be shared (and must be, in symmetric cryptography, where the term is correctly used) private keys are not shared.

Because there is no shared secret in Public Key Cryptography, the key distribution problem that has bedevilled symmetric key cryptography is transmuted into the problem of ensuring the integrity of published information concerning the owner of the matching private key. No confidential channel between communicating parties is needed to perform a key exchange.

As illustrated in Figure 1, if Alice wants to send Bob a secret message (Alice and Bob are the crypto junkies of the literature), she encrypts it using Bob's public key, which we assume everybody knows. Only Bob can decrypt this, since only he knows the value of the corresponding private key. In fact anyone in the world can converse confidentially with Bob in this way without any previous contact.

using Alice's public key, but since Alice alone knows the private key it must have been her who encrypted it. Such an encryption with a private key is known as a **Digital Signature**, and it has some of the properties of a real handwritten signature on a piece of paper. Because no-one knows Alice's private key except Alice, Bob could convince a third party that it really was Alice who signed the data. If the circumstances surrounding the signing were set up carefully, it would be hard for Alice to convince the third party that she did not do it, i.e. for her to repudiate the signature. This built-in property of **non-repudiation** is unique to Public Key Cryptography.

Although the public key of a person is not in any way secret, it is important that anyone depending on such a key really knows who the owner is of

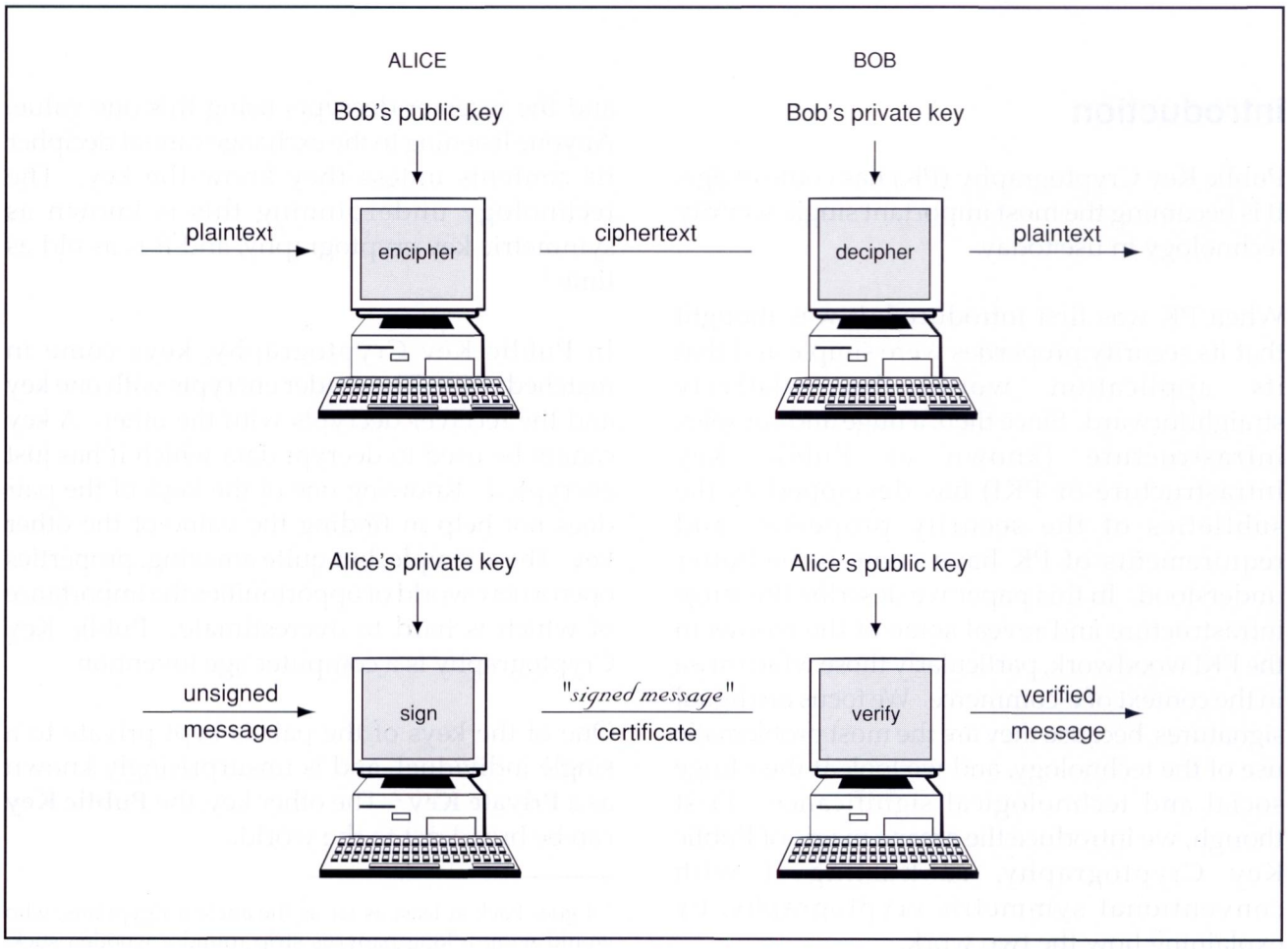


Figure 1: Public Key Encryption

If Alice wants to send Bob something that she wants Bob to know for certain came from her, she can encrypt it, or in practice encrypt a digest of it, using her own private key. Bob, or for that matter anyone else in the world, can decrypt it

the corresponding private key. If Alice is to encrypt something for Bob, she really does need to know that the public key she uses to perform the encryption belongs to Bob. To provide this guarantee of ownership a **Public Key Certificate**

is used. This is an electronic statement, digitally signed by a known and trusted authority, a **Certification Authority** or CA, using its own private key, naming the owner of a particular public key.[3] The owner is referred to as the **Subject** of the certificate. A certificate also contains a mountain of supplementary information, one of the more important items being an expiry date beyond which the certificate is no longer valid. The form of such a certificate has been standardised and it is known as an X.509 certificate after the number of the ITU standard that defined it [ITU, 1999].

To verify the signature on a certificate, we need to use the public key of the CA that signed it, so the problem of knowing to whom a public key belongs recurses: how do we know that this second public key belongs to the real CA? To help solve this problem, public keys belonging to CAs can themselves be certified by other certificates, and so on, giving rise to certificate chains. In practice, the logic of following certificate chains can get very complicated, giving rise to a host of consistency problems. Figure 2 shows an example of a certificate chain. In a typical e-commerce application, Alice[4] would validate Bob's public key, either by following the chain down from X through Y and Z, shown as a red line, or, if she knows of its existence, by taking the short cut shown as the dashed blue line directly across to Z by means of what is called a **Cross Certificate**, which leaps across the hierarchy chasm, reducing the chain length. The logic goes something like this: using the hierarchic method, Alice has had X's public key built into her system, where X is a CA she trusts.[5] X has delegated a trust for signing certificates to Y, which in turn has delegated it to Z. Z has signed Bob's certificate, which Alice can now verify. Using the short cut method, Alice has had built into her system the public key of V, the CA directly above her in the hierarchy. A certificate for Z's public key, signed by V, exists. Bob's certificate can now be validated using Z's public key. Other certificates shown in the figure are used by Bob to validate Alice's public key.

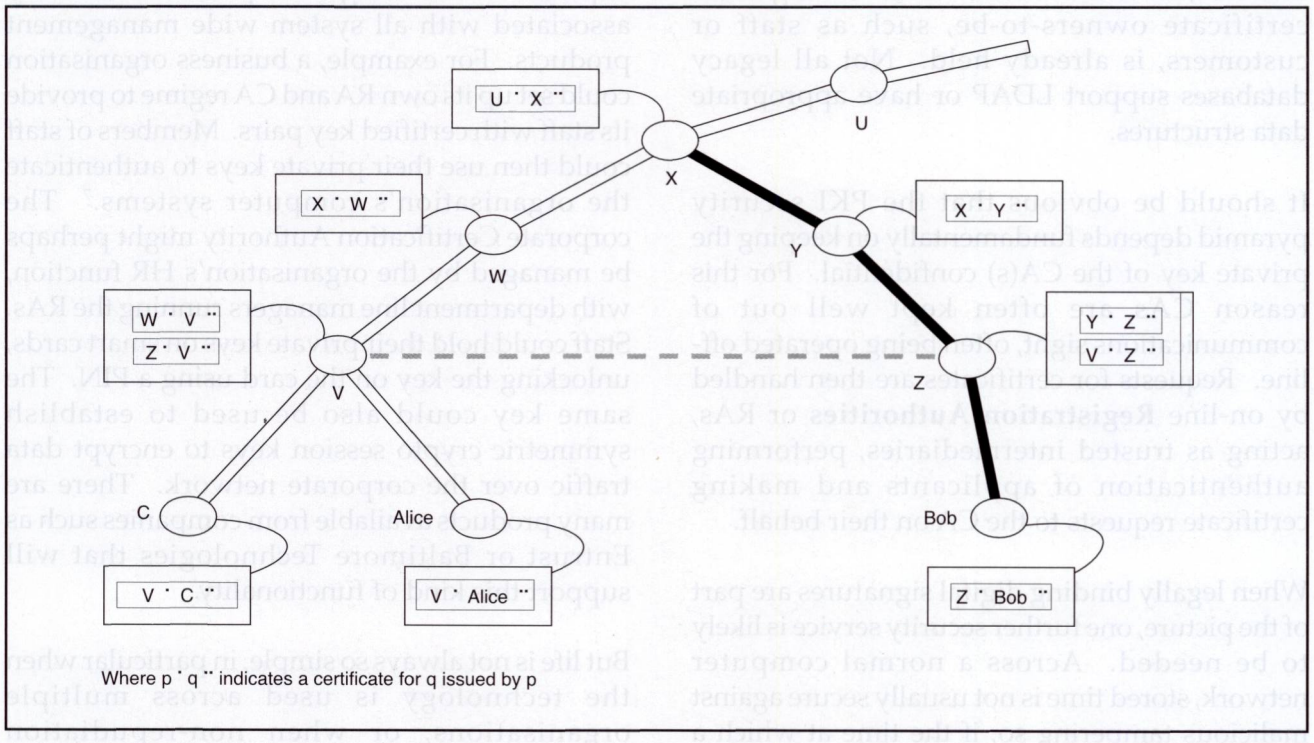


Figure 2: Public Key Certification

[3] More significantly, it is the owner of the private key corresponding to the public key in the certificate that is identified by implication.

[4] We speak about "Alice" but really mean some automated software acting for Alice.

[5] Or more likely in practice, Alice has no idea whether to trust it or not, but accepts it simply because its public key has been built into her software.

In the end though, to break the recursion, a user of public keys needs to know at least one public key that is guaranteed by some other means to belong to a known and trusted CA. In an Internet context, this root CA public key value is built into the browser. This gives rise to a serious commercial issue: it is browser suppliers such as Microsoft and AOL/Netscape who dictate which commercial CAs' public keys are chosen for this important position. A browser supplier can make or break a global commercial CA by including or excluding its public key from their browser.

In a practical e-commerce application, a large number of public key certificates will need to be stored somewhere where they can be readily accessed. This **Certificate Repository** does not need to be particularly secure, since certificates are self protecting, so a standard commercial X.500 style directory can be used. The PKI standards specify LDAP[6] as the standard protocol. It should be noted, though, that if a commercial organisation is adding a PKI to a corporate network, the certificate repository will probably need to be part of a legacy corporate database, since this is where data relating to the certificate owners-to-be, such as staff or customers, is already held. Not all legacy databases support LDAP or have appropriate data structures.

It should be obvious that the PKI security pyramid depends fundamentally on keeping the private key of the CA(s) confidential. For this reason CAs are often kept well out of communications sight, often being operated off-line. Requests for certificates are then handled by on-line **Registration Authorities** or RAs, acting as trusted intermediaries, performing authentication of applicants and making certificate requests to the CA on their behalf.

When legally binding digital signatures are part of the picture, one further security service is likely to be needed. Across a normal computer network, stored time is not usually secure against malicious tampering so, if the time at which a document is signed is important, a trusted **Time Stamping Service** is needed to accept a signed document, append a guaranteed and trusted current time value and sign the result.

[6] Lightweight Directory Access Protocol

Finally, there is the thorny problem of **Revocation**. There are many reasons why a certificate might need to be revoked and we identify some of these later. Two forms of solution have arisen to support this requirement: **Certificate Revocation Lists**, or **CRL**s, and direct status checking using an **On-line Certificate Status Protocol** (**OCSP**). A CRL is a list of certificates that have been revoked, signed by the CA that issued the certificates concerned. When validating a given certificate, the evaluator checks the appropriate CRL to make sure that the certificate is not identified there. OCSP is a more immediate check using an on-line security server, trusted by the CAs it supports to correctly respond to individual certificate status queries. There are serious issues surrounding revocation which will be discussed later.

How easy is all this to implement?

There are many uses of a PKI which are relatively straightforward to implement and manage. A PKI solution which is applied internally within a single organisation should pose only the deployment and integration problems normally associated with all system wide management products. For example, a business organisation could set up its own RA and CA regime to provide its staff with certified key pairs. Members of staff could then use their private keys to authenticate the organisation's computer systems.[7] The corporate Certification Authority might perhaps be managed by the organisation's HR function, with department line managers running the RAs. Staff could hold their private keys on smart cards, unlocking the key on the card using a PIN. The same key could also be used to establish symmetric crypto session keys to encrypt data traffic over the corporate network. There are many products available from companies such as Entrust or Baltimore Technologies that will support this kind of functionality.

But life is not always so simple, in particular when the technology is used across multiple organisations, or when non-repudiation properties are required. Unfortunately, it is these forms of PKI which will be most needed by electronic commerce. So now the bad news...

[7] Authentication using a private key usually takes the form of a signed response to a random challenge.

Some complications

What is in a name?

When depending on an externally provided public key certificate, its limitations must be understood. Bob Blakley, an American security expert from IBM, once memorably described it as being a sort of marriage certificate:

"By the power vested in me,[8] I now declare this text string and this bit string: 'name' and 'key'. What RSA[9] has joined, let no man put asunder."[10]

The bit string is of course the public key, which has direct significance in computer terms — indeed it has no meaning outside computer systems. But the text string needs a real world context to give it significance. A guarantee that key X belongs to a text string means very little in itself. The statement only acquires value when the following conditions are met.

1. The text string is unmistakably understandable as the name of a single accountable real world entity such as a human individual, a business organisation or other body corporate.

   The text string "Alice" would not pass this test — the name is hardly unique. "MICROS0FT"[11] is not the same as "MICROSOFT" but could be used to mislead.

2. The name has a context, either explicitly stated or implied.

   If "Tom Parker" is to be trusted to sign an electronic purchase order on behalf of ICL, the verifier of his signed orders will find it of little value to know that his golf club Certification Authority signed his public key certificate. The golf club's CA is trusted to name him, by implication as a member of the club, but there is no implication of any relationship with ICL, nor any authority to sign purchase orders.

3. It is clear who is going to be accountable.

   The verifier needs to know whom to sue, and for how much they can be made accountable if the signatory repudiates the signature. Unfortunately accountability is a thorny problem; commercial Certification Authorities go out of their way to avoid any accountability for consequential loss.

To these issues we must add the further question: can the naming system in the certificates be successfully integrated with legacy naming systems used for such things as access authorisation?

So where there is apparently a certificate merely linking a name to a key, its true value is related to the surrounding context. It could be argued, therefore, that for electronic commerce, a certificate that enriches the context and makes it more explicit would be of more value; such a certificate is known as an **Attribute Certificate**. Attribute Certificates are currently being standardised in the IETF [IETF, 2000], although these are not yet in general use.[12]

Attribute Certificates

A certificate that says, "the owner of the private key corresponding to the public key in this certificate, whoever it is, is good for financial liabilities up to a total of £1,000 and I, the CA, will be accountable if the owner fails to pay," would be more directly useful in an e-commerce context. It has parallels with the current cheque

[8] the Certification Authority.

[9] RSA is the name of the public key algorithm most used in PKI implementations.

[10] One could add of course "..until a divorce by revocation or widowhood by expiry".

[11] With a zero between the "S" and the "F".

[12] There are isolated exceptions. One is in the e-commerce SET [SET, 2000] protocol, which defines a certificate containing a credit card number.

guarantee card liability situation, which has worked well for several years now. Contrast this with what current commercial Certification Authorities are saying about the certificates they produce; some of the more jaundiced members of the IT community might roughly paraphrase this as (see Figure 3):

> "At the time it was requested, I, the CA, went through my published process to verify the applicant's identity, and only the identity. I do not know the applicant any more than the man on the Clapham omnibus does. I do not know whether the applicant is trustworthy, or even, if it is a company, what its business is. I will not necessarily find out if the private key associated with the certificate is stolen and is being used by an impostor, nor do I necessarily know whether the applicant has since changed name or has ceased to exist (but if I do learn either of these things, I'll publish what I know. Sadly though, the software you are using is unlikely to be able to make reliable use of this information). The procedure I went through may or may not be a thorough one, depending on the type of certificate this is. Not all software understands these different certificate types, and you probably won't know whether it does or not. Enjoy!
>
> P.S. Do not try to sue me unless you can prove that I didn't follow my published certificate issue process. You will not win..
>
> 

Figure 3: Attribute Certificate!

The CAs who sign Attribute Certificates will need to be different in character from the commercial CAs that dominate the market today. Attribute Certificate CAs would need an independent real-world knowledge of the attributes that are associated with their subscribers. They are likely to be directly involved in the business context associated with the attributes. There will consequently be more of them and they will be smaller. If they are to fulfil their promise, their acceptance of liability will need to be higher, with the consequent raising of the cost of certification, which would then become akin to an insurance premium.

In the real world, people operate within a wide variety of commercial contexts, so at first sight it would seem that, if Attribute Certificates were to become widely used, people would need more keys and certificates than if they just had normal identity based certificates. However, the difference is an illusion. If identity certificates were being used properly, a similarly large number of them would be needed, each with a different implied context, signed by the different CAs that should be operating within the contexts implied. It is only because the liability model has not yet been properly taken care of that there is this superficial difference.

A pure Attribute Certificate could be anonymous to the verifier, with only the CA knowing the owner's identity but, if it is not, Data Protection Act (DPA) considerations would come into force — the more interesting the attributes in a certificate, the more concerned the owner might be about its confidentiality. Once the spectre of certificate confidentiality appears, the PKI solution starts to break down, since the essential quality that a certificate is totally self-protecting is lost. This could become a significant issue. On the other hand, we must not forget that, as already argued, even an identity certificate carries implied attribute values, and one might ask whether the DPA covers the implied loss of privacy which arises from this. These DPA issues have yet to be resolved.

A disadvantage of Attribute Certificates is that the information in them tends to be more volatile than identity certificates, so that their useful lives are consequently likely to be shorter than those of identity certificates, making certificate management more onerous. For example, a certificate containing someone's job role will be more short-lived than an identity certificate, even if the identity certificate is one that has been signed by the corporate CA, with the contextual implications of being a member of staff. People leave organisations less frequently than they change job within them. Some system security infrastructures[13] make use of mayfly-like Attribute Certificates that live for only a short user session — they are born at log-on, authorise access to resources, then die at log-off.

[13] For example the European SESAME technology [Sesame, 2000].

Certificate Revocation

Revocation is the Achilles' Heel of PKI. It is the crumbling at the edges of PKI. It is a minefield. In the beginning, one of the great advantages of PKI was that, in principle, it required no on-line security services during normal running. Given that a signer sends, with the signed data, all the certificates in the certificate chain needed to verify the signature and, given that a top level CA public key value that could act as a root key for verification of the certificates sent is built in at the verifier's computer, all the security checks could be done locally by the verifier.[14] If the top level CA is a globally recognised one, even a globally sized security solution is not a problem, since all the security data needed could be carried as part of the participants' communications traffic. Unfortunately though, even a validly signed and unexpired certificate may need to be rendered invalid, i.e. revoked. There are many reasons for this. For example:

- there may have been a private key compromise or a smart card containing the private key may have been lost or stolen
- the key owner who is named in the certificate may have changed name or may be an organisation that has ceased operating
- if the key holder named is itself a CA, it may have been compromised
- if the certificate has any implications of rights of any kind guaranteed by the signing CA, the key owner may have fallen from grace or the CA may no longer be prepared to honour its commitments in relation to the certificate for other business reasons.

But how is revocation to be carried out? One solution is the CRL, but it is difficult for a verifier to ensure that it has the latest CRL from the CA. Different ways of dealing with this problem have been proposed, but the simplest is that the CA should issue CRLs at regular predetermined times. At each "checkpoint" time, even if no revocations have been made since the previous time, the previous CRL is reissued with a new time-stamp on it — it is as important to be certain that no additional revocations have taken place as it is to learn of any that have. CRLs are not a perfect solution since by their nature they cannot be one hundred percent up-to-date.

Recent experience in designing practical PK infrastructures has identified a host of real world implementation complications with CRLs, so that their semantics and the semantics of the individual CRL entries, are becoming ever more complex. Some of the information in the current CRL standard[15] will illustrate this:

- **Revocation reason** — see above for an example list, though not all of the possible reasons are in the standard
- **Hold instruction code** — a certificate may not be permanently revoked, it may be just on hold. If so, the CRL may contain an indication of what the certificate verifier is to do in connection with the hold. Examples given in the standards are codes to indicate: "please communicate with the CA" or "repossess the user's smart card"
- **Invalidity date** — the date at which the certificate is to be considered invalid (this may be before the date of issue of the CRL). This is advisory, since a key holder may advise that the date of loss was before it was actually lost in order to repudiate a signature made after that date but before the loss was advised to the CA
- **Delta-CRL indicator** — needed when CRLs get very large. Instead of issuing a full CRL every period, only changes, or "deltas" are issued
- **CRL Issuing Point** — information in a certificate indicating where to find the CRL that might have revoked it.

These, and other more esoteric fields not listed here, are poorly and inconsistently supported in commercial software, so perhaps the answer is to use On-line Certificate Status requests to an

[14] It has to be said though, that certificates can get very large, and to send all the certificates needed for verification with every transmission could be prohibitively costly in communications terms.

[15] This is the same standard as the certificate standard, i.e. X.509, but the current version of the CRL part is V2.

On-line trusted Certificate Status Server. The server could either itself contain the complex logic needed to search for and analyse the CRLs it needs,[16] or it could provide a direct service by being kept up-to-date in real time by the CAs it supports. On-line Certificate Status Servers, however, have the big disadvantage that an on-line check needs to be made, losing much of the advantage of using a PKI. Indeed, in the limit, once a server exists which is updated in real time from a CA, why have certificates at all? Why not simply ask, "I have a data item here, signed using private key K, tell me all you know about the signer?"

Finally there is the question of when to revoke a suspect certificate. Should the timing be decided by the CA or by those who are relying on the signature? There are e-commerce applications[17] in which large numbers of signed messages lie in the system for long periods of time, perhaps weeks, before being verified. Suppose a security incident occurs on the computer that performs the signings and the management are concerned that the private signing key may have been compromised, but are not sure. Do they immediately revoke the signing key? If so, a large number of perfectly valid signed messages in the system will, over the coming weeks, start to fail their verification. The decision to revoke needs to be carefully timed so that the known and certain cost of recovering from the failed verifications of valid messages is balanced against the possible cost of invalid messages being wrongly accepted.

Standards — the interworking problem

Of course there are the usual host of standards (Figure 4) [IETF, 2000], but as Peter Gutmann says in the X.509 Style Guide [Gutmann, 1999]:

> *"Anyone who has had to work with X.509 has probably experienced what can best be described as ISO water torture, which involves ploughing through all sorts of ISO, ANSI, ITU, and IETF standards, amendments, meeting notes, draft standards, committee drafts, working drafts, and other work-in-progress documents, some of which are best understood when held upside-down in front of a mirror (this has led to people trading hard-to-find object identifiers and ASN.1 definitions like baseball cards — 'I'll swap you the OID for triple DES in exchange for the latest CRL extensions')."*

Figure 4: Usual Host of Standards

Towards the end of 1999, the Universal Postal Union, a United Nations organisation of national post offices, conducted an experiment. They took six commercial PKI products, all claiming to support the X.509 V3 standard, and five secure mail agent products, all claiming to support the secure S/MIME V2 protocol 18. They developed interworking tests, such as: get a public key certificate using PKI product X, sign a mail item using the corresponding private key secure mail client Y, mail it and the certificate to mail client Z where the signature should be verifiable.

Of the 267 tests attempted, 5 succeeded and 262 failed.

Industry groupings have been set up to try and resolve this problem. Some of the leading ones are:

- The PKI forum, which says that it is "an international, not-for-profit, multi-vendor

[16] Though it has to be said that this is not the IETF's intended way that OCSP will work.

[17] ICL Pathway is one of them, albeit a rather specialised form of e-commerce.

[18] S/MIME is a protocol standard for formatting cryptographically protected mail items. In particular it dictates the format of signed mail items [S/MIME, 2000].

alliance whose purpose is to accelerate the adoption and use of Public-Key Infrastructure (PKI) and PKI-based products and services" [PKI Forum, 2000]

- "Trusted Infrastructure for Europe" (TIE) being led by ICL, with Baltimore Technologies, IBM, the Post Office and Shell as other leading members. TIE will also be conducting practical interworking trials and demonstrations

- "Identrus" — a group of leading banks (from all over the world) who have agreed on technical and certification practice standards, and will act as authorities with which commercial organisations can register. Participating organisations trust the Identrus banks to have checked the business reliability of organisations registered with them, and so do not have to do this themselves. This looks like being an important and successful initiative. Identrus plans to go live in early 2000 [Identrus, 2000].

There are other groups emerging too, with the consequent danger of multiple different group interworking implementations not then interworking with each other!

Digital Signature Legislation

In the UK, the Electronic Communications Act has progressed through the House of Commons. Subject to certain constraints, it makes a digital signature as legally binding as a pen and paper signature. This is a laudable achievement which will save millions of pounds by simplifying commercial processes. It will no longer be necessary to print out paper copies of legally significant computer transactions for them to be given a physical signature; i.e. the whole business process will be able to remain electronic.

The act's progress was not always smooth. Early drafts of what was then known as the Electronic Communications Bill were controversial because the Government tried to carry on its back legislation relating to enforceable decryption. Succumbing to the huge amount of highly vocal opposition from most sectors of the computer industry, the Government moved this part of the legislation into a separate bill. There remain, however, some significant issues. Although digital signatures seem on the surface to be very similar to paper ones, there are some major differences:

- experience of all of the legal wrinkles that surround the performing of paper signatures has been developed over a thousand or more years. People are familiar with the precautions that need to be taken in order to avoid being spoofed into false signatures; there is a vast history of case law. Digital signatures are new, highly technical and not understood in the slightest by non-specialists, who would be unlikely to be able to defend themselves against a fraud attack

- when a paper signature is made, it is clear that the signer intended to make it. If a signature is to be legally binding, the signer's intent to sign in this way must be clearly demonstrable. In the electronic world, the agent who makes the signature on behalf of the signer is not a simple pen, but a complex computer system which is quite capable of making a signature without the signer even knowing it, let alone intending it

- the means of making a paper signature cannot be stolen — it is the signer's brain, arm and hand. A private key can be stolen.

Issues will also arise because of the many steps involved in the process of generating the means of performing a digital signature and applying it, any of which can (and often will) go wrong, even if no malice is intended. In the event of a dispute, a court of law will have to be convinced that the processes involved have retained their integrity and have performed in a sufficiently reliable and secure manner. Sadly, a court will not have the technical know-how to make a sensible judgement without bringing in expensive technical advice. Arguments are therefore being made for a standard set of guidelines to be drawn up for courts of law.

The legislative position is complicated by the international nature of electronic trade and the different and incompatible digital signature laws which have been passed in different countries. Even within Europe, the European Union draft directive on Electronic signatures is incompatible with many of the individual member states. For example, the EU guidelines forbid a member

country mandating that Certification Authorities be subject to government licence, but Germany and Italy currently do require such a licence if a digital signature verification using a certificate issued by them is to be binding. The EU guidelines will also require Certification Authorities to accept some degree of consequential liability for actions performed under the guarantee of their certificates, the actual liability limit being present in or referenced from the contents of the certificate itself. It will be interesting to watch the commercial CA community's reaction to this.

Conclusions

Public Key Cryptography is an immensely valuable technology. It is having a huge positive impact on the viability of e-commerce across the Internet. As its use becomes more prevalent the need for a substantial PKI will grow ever stronger.[19] Unfortunately, even though the PKI infrastructure is well developed, it has not yet been fully fired in the implementation furnace of e-commerce. The science is reasonably mature, but the engineering is not. There are substantial practical issues surrounding revocation and the trust relationships which exist between the different trusted third parties involved. Interworking has not been well achieved, even though business groupings have been formed to try and solve this problem. The societal impact on people and businesses of depending on legally binding digital signatures has hardly been studied. Governments, undeterred, are nevertheless passing differing legislation to allow signatures to be binding.

As Jan L.A. van de Shepscheut once said, "In theory there's no difference between theory and practice, but in practice there is!"

Epilogue

The author has not had the space to cover the vast world of PKI. Digital signature legislation has not been described in any detail, nor has any attempt been made to discuss the impact on e-commerce security of cryptographic export restrictions. The more detailed semantics of some of the protocols and data structures in day-to-day use have not been covered and protocols such as SSL and SET have hardly been mentioned. No attempt has been made to describe how PKI infrastructures fit in different ways within different e-commerce models and the close links between PKI and Virtual Private Networks have been ignored.

It is to be hoped that these missing topics will be discussed in detail in future issues of the Systems Journal.

Bibliography

GUTMAN, "X.509 Style Guide" Peter Gutmann, September 1999, pgut001@cs. auckland.ac.nz. http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt.

IDENTRUS, The Identrus Web site is at: www.identrus.com/.

IETF, The standards defining the PKI that should underpin the use of Public Key Certificates are being developed by the IETF's PKIX group. They are available free from the Internet at: http://www.ietf.org/html.charters/pkix-charter.html

ITU, "The Directory — Authentication Framework", CCITT (now ITU) Recommendation X.509. The version of the certificate standard supported by most PKI products is V3. It is available for purchase via: http://www.itu.int/itudoc/itu-t/rec/x/x500up/x509.html.

PKI FORUM, The PKI forum's Web site is at: www.pkiforum.org/

SESAME, The SESAME Web site is at: www.cosic.esat.kuleuven.ac.be/sesame/

SET, SET (Secure Electronic Transactions) is an e-commerce PKI based transaction protocol supported by MasterCard, Visa, Microsoft and Netscape, among others, but despite the support of these "big guns", has had a rather lukewarm response from the rest of industry because of its complexity and performance problems. SET

[19] Market research firm, International Data Corporation, believes the PKI market will take flight and revenues will soar. IDC say that from a base of only $122.7 million in 1998, worldwide total PKI revenues will reach a high-flying $1.3 billion by 2003, see www.idcresearch.com/Data/Internet/Content/NET122199PR.htm.

specifications can be obtained from the Internet at http://www.visa.com/cgi-bin/vee/nt/ecomm/main.html.

S/MIME, The S/MIME working group page is at www.imc.org/ietf-smime/

Biography

Tom Parker graduated from Cambridge University in 1963. In 1971 he joined ICL as an operating system designer on VME, and in 1978 started to specialise in computer security. During the mid 1980s he was responsible for the inception and much of the design of the VME High Security Option, which was certified by the British Government as the UK equivalent of B1 on the American DoD Security Evaluation Criteria scale. During the late 1980s he was the ICL Security Architect in the European consortium developing the innovative SESAME distributed security technology. In 1991 he was elected an ICL Fellow. He has a special interest in cryptography and the security of distributed systems, and was the Security Architect responsible for the ICL Pathway crypto infrastructure, which makes extensive use of Public Key Cryptography. In October 1999 Tom retired from ICL and he is now an independent consultant. He is married with two grown up children.

Ec^x — An Award Winning System

Dick Emery

ICL, Kidsgrove, UK

Abstract

Multivendor Computing, formerly trading as Technology plc, is a thriving division of ICL. It is a major supplier to the UK market of PC equipment from Fujitsu, Compaq, SUN, Toshiba and many others. It specialises in creating close relationships with the procurement departments of large corporate customers. In 1995 Multivendor Computing embarked upon a project to network itself to its major customers. The result was the system known as Ec^x which provided much closer integration between the internal systems and processes of both customers and Multivendor Computing. Over the ensuing years Ec^x has grown and prospered. Its customer users have reduced costs and gained in convenience and efficiency. Multivendor Computing has also expanded and won new customers as a result. This paper describes the Ec^x system, its achievements and its lessons.

Multivendor Computing

In 1990, ICL acquired Technology plc, the successful PC re-seller. At that time, Technology plc was a major outlet for ICL-branded PCs and associated equipment. It was also a significant outlet for equipment from other manufacturers such as SUN, Toshiba and IBM. For some years, Technology plc continued to trade under its own name as a part of the ICL Group. In 1998, Technology plc became fully absorbed into ICL, changing its name to Multivendor Computing, as an ICL division. Meanwhile ICL, being a wholly owned member of the Fujitsu Group, rationalised its PC design and manufacturing interests with those of Fujitsu and ceased to design and manufacture PCs in its own name. Multivendor Computing thereby became a major supplier of Fujitsu equipment to, primarily, the UK market.

Multivendor Computing is today the largest UK supplier of Fujitsu products. It is also a top Compaq re-seller in the UK and is the UK's only SUN Master Reseller. It is a major outlet for Microsoft software as well as PC and ancillary equipment from Toshiba, HP, IBM, Epson and a variety of other well-known suppliers.

Ec^x Overview

Ec^x is a corporate purchasing system. In effect, it connects Multivendor Computing's sales and fulfilment systems to those individuals in major corporate customers who wish to procure PCs, ancillary equipment, accessories, software, services and mobile telephony. The connection threads itself through each customer's procurement and approval process in order to automate as many steps as possible. To achieve this, Ec^x comprises three major sub-systems:

- an **electronic catalogue** which exposes what is available for purchase in a manner tailored to the purchaser
- an **order management** system which pilots the participating individuals correctly through the creation, confirmation and progressing of orders
- a **transaction gateway** which securely and rapidly accepts in-bound EDI-compliant electronic orders from customers and forwards requests for stock replenishment to suppliers in order to allow fulfilment of confirmed orders.

Each corporate customer chooses how it wishes to interface to the full collection of facilities on offer. For example, those who wish to confirm their orders using facsimile or surface post can do so. Connection may be across the Internet or by private circuits. In some cases, the corporate customer may mount its catalogue on its intranet behind its fire wall and issue orders on-line. The system is constructed to accommodate any sensible combination which suits the individual corporate customer.

The descriptions in the following sections assume that the corporate customer uses the full facilities across the Internet unless the text indicates otherwise.

Brief History

Multivendor Computing, then Technology plc operating at arm's length from the rest of ICL, first became interested in implementing electronic connection to its major customers in 1995. The management had assumed that this would result in reduced costs for both themselves and their customers. They viewed this as a vital factor, operating as they did, and still do, in a sector of the IT industry renowned for its thin margins. They knew that financial success relied heavily on the strict control of all costs and that electronic connection held the promise of significant savings. Some even viewed a move in the electronic direction as vital for continued survival let alone prosperity.

The early days of the resulting project were not especially happy. Technology plc management chose a non-ICL supplier for what they had in mind and soon became disillusioned by lack of progress. They decided to take advice from their ICL colleagues and hence made the decision to move the project in-house. The project then moved forward quickly in order to meet the original deadlines despite the intervening delay.

The early concept for the system exploited the regular issue of CD-ROMs to customers as the vehicle for communicating product catalogues. However, the advancing popularity of the Internet soon altered perceptions and a more comprehensive system emerged which now uses the Internet and equivalent intranet technology. The CD-ROM option is now historic.

In early 1997, Ec^x took on its first customer. The initial feedback from the early customers was rewardingly enthusiastic. Those responsible for the operation and implementation of Ec^x decided to enter for the 1997 Information Management Awards, supported by amongst others Deloitte & Touche. To no one's surprise at ICL, Ec^x duly won the award for Project of the Year in the electronic commerce and communications category. ICL's internal annual Engineering Conference at about the same time also saw Ec^x win a major commendation for outstanding achievement.

During the past three years of successful operation, the number of customers has risen to over 250, of which over half are regularly active users.[1] The less active users are typically more recent joiners who are evaluating how best to use Ec^x or are adapting their ways of working towards it. The customer roll includes familiar names from government, banking, manufacturing and the utilities. From its UK-only beginnings, Ec^x now extends into a growing list of European countries, at the time of writing reaching ten, which account for about 10% of its total revenue. The system has now happily processed orders with values from a few hundred through to well over a million euros.

The number of live product lines on offer exceeds 38,000. Of these, 7,500 account for 90% of the shipped items. All the product lines relate to volume IT and mobile telephony products. In light of the rapid rate of obsolescence in this class of product, maintaining the currency of the information stored about so many product lines constitutes a significant challenge. Hence Multivendor Computing concentrates on providing a greater information richness for the 7,500 core products than for the remaining slower moving ones.

Customer Roles

Ec^x sets out to match what actually happens in the typical corporate purchase process. We can view these as a set of roles (rather than human individuals). A typical set is shown with their responsibilities in Table 1.

In this generic model, the procurement manager negotiates purchase arrangements with an approved supplier — popularly known as a frame agreement. A user selects what products he wants in order to do his or her job. The line manager confirms that this is indeed appropriate. The technical authority and financial manager confirm conformance to standards, budgets and processes. The procurement officer formally confirms an order, containing what the user selected, to the approved supplier, then monitors

[1] Some very large customer organisations may operate as a number of independent buyers for the purposes of Ec^x. The numbers given refer to the number of independent buying organisations owning their own catalogues (see later section — Catalogues) and not the numbers of "bodies corporate".

Role	Responsible for . . .
User	Identifying what equipment is wanted for his or her role
Line Manager	Overseeing effective procurement and budget deployment
Technical Authority	Ensuring that PC procurements are technically sound and conform to the appropriate technical standards and policies
Financial Manager	Assuring the integrity of assets, budgets and cash flows
Procurement Manager	Negotiating contractual terms and prices with selected suppliers
Procurement Officer	Processing authorised orders in line with negotiated arrangements

Table 1: Roles and Responsibilities

progress to delivery to the user and finally releases payment to the supplier.

Ec^x in its complete form provides support in a flexible manner for the execution of all these steps as responsibility for a particular purchase moves from one role to the next. In this sense it is very different from an Internet merchant site aimed at consumers where one individual — the consumer — plays all the roles. In the corporate world, roles such as these are normally quite distinct and there is a well-orchestrated process, supported by documentation and messaging, which links the role players. Ec^x is built to satisfy the corporate world.

Catalogues

At the heart of Ec^x lies the concept of a catalogue. Each corporate customer (or independent buying unit thereof) has its own catalogue. A catalogue contains the products and associated purchasing terms which apply to its "owning" corporate customer. Each catalogue is in principle unique to its corporate customer. Staff from a corporate customer can view only the catalogue relating to their employer.

Hence a catalogue embodies the purchasing agreement negotiated by the procurement manager with Multivendor Computing. It includes the approved selection of products to support the customer's IT policies. For example, notebook PCs might have to be selected from one of three Fujitsu models. It includes the negotiated list price discount and delivery period — a lower price may accompany a longer delivery period. The recruitment of a new corporate customer triggers the creation of a new catalogue.

All the catalogues are virtual in the sense that they are merely views of the same database suitably filtered and augmented by personalising data. The source product database contains all Multivendor Computing's product lines with the expected complement of part numbers, categorisations, descriptions, specifications, list prices and the like, plus links to pictures and to brochure material where these are available.

Access Modes

Customer staff may access Ec^x through one of two interfaces. They may use either a standard browser or custom, client-end software known as Order Manager. The two interfaces offer different routes to essentially the same collection of products, information and facilities, given that the user has the appropriate privileges. In effect, Order Manager locates the same functionality at the client end in custom code as the browser does at the server end in a module known as Server Order Manager. Client-end Order Manager, however, can perform some editing and manipulation tasks which are difficult, and perhaps impossible, to offer from a standard browser. But the browser interface provides the important benefit of not requiring any prior downloading and installation of custom software, this being important for the induction of large numbers of low intensity users.

Those who use the browser interface also have the option of by-passing Server Order Manager and accessing their catalogues in a mode which is similar to the one found on many consumer merchant sites. This is particularly suited to users who come to Ec^x only very occasionally. It is this interface which is described in a later section (User View).

Development Issues

No two corporate customers have identical requirements and processes. As a matter of policy, Multivendor Computing attempts to meet the special requirements of each new customer. This creates significant potential challenges for the developers of Ec^x. In particular, there is a constant trickle feed of new software versions which have to co-exist happily with previous versions as well as interfacing to one set of facilities on the Ec^x Web server operated by Multivendor Computing. The solution adopted for the Ec^x system involves three co-operating elements:

- All communication proceeds through object types. Once defined, an object of a given type never changes. If an existing object requires extension or amendment for the implementation of a new facility then a new object is defined. An old object type disappears only when it falls out of use

- The latest version of Order Manager is always available for downloading from the Ec^x Web site. This means that all customers can enjoy new facilities as they are implemented. However, there is no compulsion and hence no need to monitor upgrades. Those who use Server Order Manager automatically gain access to the latest extant version because there is only one version on the Ec^x Web site

- Order Manager's detailed behaviour is controlled by an ".INI" file. Whenever a user loads an Order Manager, that instance of Order Manager downloads the current version of the ".INI" file appropriate to the customer as part of its handshake with the Ec^x Web server. Every instance of Order Manager has defaults for all the ".INI" settings which it "knows" and ignores ".INI" settings which it does not "know". Thus the ".INI" file and the version of Order Manager are not inherently version sensitive to each other.

To date, this approach has proved completely successful in allowing continual enhancement to the Ec^x system without inconveniencing existing customers.

User View

The user (as defined in the section on Customer Roles) accesses Ec^x via a normal browser routed through to Multivendor Computing's Web site. The user has to know no more than the URL for the Ec^x welcome page[2] and he obtains this from colleagues, internal distribution or whatever is appropriate for the given corporate customer. The welcome page asks for an identity and associated password since the Ec^x site is not open to the public. In any case, Ec^x has to ensure that only the authorised staff of a given customer are able to see the negotiated prices available to that customer.

The given identity, providing that it is accompanied by a valid password, corresponds to a catalogue (as defined in the earlier section on catalogues) thereby allowing Ec^x to present the appropriate list of selectable products, prices and delivery terms to the user. There is an option to visit as a guest and this invokes a maximum, unselected, default catalogue which quotes standard list prices without discounting.

The user then has access to the facilities associated with the catalogue. The main feature is that of browsing. Every catalogue is organised hierarchically with the highest level separating the entries into "Accessories", "Hardware", "Nets & Comms", "Peripherals", "Services", "Software", "Consumables" and "Telephony". These eight categories always appear as clickable tabs across the top of the screen, making jumps between them very easy.[3] Each category when first entered offers a list of clickable sub-categories. Each sub-category may similarly offer a further list of clickable sub-categories on the way down the hierarchy finally reaching a list of product lines.

The pages which show product lines have three zones. The upper zone continues to show the

[2] This is currently https://www.iclmc.com/ec/login/login.asp.

[3] Some catalogues show fewer categories because they include no products within the omitted categories.

eight top-level categories together with further tabs which indicate the route from the top of the hierarchy to the current position. An example could be "Software/Desktop Apps Shrinkwrap/Suites/Microsoft" whose product lines are the many variants of Works and Office. The middle zone gives expanded details of the currently selected product line, covering title, description, price and delivery lead time. There may also be a picture. Completing the middle zone is a set of six function buttons for ordering, displaying brochure information, searching and so on. The lower zone is a scrolling list of product items showing manufacturer, product name, price, lead time and part number. Each item is clickable and it is the selected item whose details appear in the middle zone.

The user employs the navigation and selection facilities, which include searching on a variety of relevant criteria, bookmarking, tab clicking and scrolling, to assemble an order for a collection of products which may include hardware, software and service elements. At this stage the collection represents a potential order since the prices and deliveries are merely indicative and there has been no authorisation of the order. When the user is satisfied with the potential order, he submits it to his local process for authorisation.

The interface as described is currently under periodic review and a pilot of a new design exists. This aims to offer the same range of information and facilities, but through an improved visual impression and more ergonomic placement of the controls.

Procurement Officer View

The browser interface described above is not well suited to the detailed duties of the procurement officer so he employs the Order Manager application, either on his PC or through Server Order Manager. This provides direct access to all the features of Ec^x without the elaboration of graphics and hierarchies which tend to impede efficiency for the frequent Ec^x user. Order Manager makes use of lists, buttons, menus and dialogue boxes in the style of most Windows applications. There is a powerful search facility which uses wild card characters and abbreviations. Order Manager's design sets out to minimise keying, thereby reducing errors and improving efficiency.

Only the procurement officer (as described in the section on Customer Roles) can commit a firm order to Multivendor Computing on behalf of the corporate customer, after first checking the potential order's technical and financial integrity. Order Manager provides the mechanisms to take the potential order through these approvals (see later section on Approval Routing). For example, the electronic record which holds the schedule of products for the order also contains essential reference information which is required by Multivendor Computing and the customer for identification and delivery purposes. The customer may augment this with extra fields and by attaching further documents which, say, justify the purchase. As the procurement officer pilots the potential order through the customer's internal process, he records each stage via Order Manager and these records are available for viewing by anyone with the necessary privilege. This may include the user who can thereby watch progress towards meeting his order request (see next section on Tracking and Reporting). Ec^x aims to provide one place in which to gather all the information relating to the potential order.

When the procurement officer has assembled the necessary authorities, he is ready to turn the potential order into an actual one. His identity and password give him this privilege. He first asks the system to insert the latest prices — these change in real time. He then checks availability — these too change in real time and some items may be available from stock while others may have to come from their manufacturers or suppliers. When he is satisfied with all the authorities, prices and availabilities, the procurement officer confirms the order using Order Manager.

Multivendor Computing's systems now process the order to completion, updating the order status as equipment is ordered with manufacturers, comes into stock and is delivered to the customer. Through Order Manager, the procurement officer and anyone else with the necessary access privilege can directly monitor the developing status of each order.

Tracking and Reporting

Some customers choose to use Ec^x solely for the benefits which accrue from using the tracking and reporting features. They consider these to

be so valuable in their own right that they alone justify the use of Ec[x]. This is understandable for large buyers who have the unenviable task of keeping track of perhaps hundreds of simultaneous orders, all in different states, while having to produce summary reports for management and regular review meetings. Only comprehensive, automatic, built-in facilities are capable of meeting these exacting needs. In this sense, the procurement officer in the large corporate customer has a unique requirement which consumers and occasional purchasers often find hard to comprehend.

For tracking purposes, Ec[x] provides via Order Manager a wide variety of ways of viewing potential, current and completed orders. For example, there is an on-screen, colour and text enhancement scheme which provides visual cues as an order's forecast completion date slips beyond the customer's requirement. There is a tailorable selection scheme sensitive to dates, values, product types and so on for highlighting key issues. Any screen selection can be printed or downloaded in a form ready for loading into a spreadsheet for further analysis. The interface for all of this is largely intuitive for the typical procurement officer and does not demand the acquisition of daunting report writer skills.

Approval Routing

The description given earlier implies that the procurement officer is responsible for routing a potential order through its authorisation steps. Many customers choose this way of operating and Ec[x] may or may not be invoked to record what happens. But Ec[x] can automate this routing too. The customer can define the names or titles of the occupants of up to sixteen authorising categories. The procurement officer can then select one name from each of the appropriate categories for the given order and "post" the order for authorisation. When all the authorisers have responded with their approval he can then release the order to Multivendor Computing. Alternatively, the system may be set to release the order automatically when all the approvals have arrived.

The approval system works by using a facility known as Approvals Manager which can sit on a Web server inside the customer's intranet or, for most customers, on the Ec[x] Web server. Approvals Manager uses e-mail facilities to communicate the order details and attachments to each authoriser and each authoriser e-mails his answer back to Approvals Manager using standard reply facilities. The procurement officer can track the accumulation of approvals on his screen as they arrive.

Approvals Manager operates according to rules in the style of a simple workflow system. The "owning" corporate can set triggers based on price, product type and the like which determine who in the management hierarchy has to grant approval. For example, a £49,000 order for Fujitsu equipment may need the responsible divisional manager's approval, but a £4,900 order for Toshiba equipment may need the IT Director's approval as well because Fujitsu is the default and Toshiba is an exception within this corporation.

Benefits

Ec[x] has the potential to benefit both the corporate customer and Multivendor Computing. At the most general level, it substantially improves the relationship between the customer and Multivendor Computing, bringing the internal systems of the two into much closer contact. In this sense, it is typical of the way in which many electronic business systems succeed in removing unnecessary cost barriers. The result for both parties is, in the Ec[x] case, faster, more efficient procurement leading to reduced total cost of ownership for the subject PC equipment, ancillaries and so on.

Although figures which prove or refute claims to success are difficult to obtain in the absence of a controlled before and after analysis, there is compelling anecdotal evidence that Ec[x] does deliver its promise. For example, most customers accept the broad claim that Ec[x] reduces the average cost of processing the typical order from £150 to £50. For large customers processing hundreds of orders every year these £100 savings accumulate to noticeable sums. There is also evidence that a customer's order cycle time is more than halved. Thus, whereas it took an average of 16 days from user initiation to order confirmation with Multivendor Computing, the period is now six days. It is difficult to ascribe a financial benefit to this reduction but it seems likely that the opportunity benefit might in some cases be substantial if the saved days make the

difference between gaining or losing a big order, or completing a vital project on time or late.

Ec^x also brings other less tangible benefits to the procurement process:

- It exploits buying patterns, making it particularly easy for procurement officers to repeat previous orders using a simple copy and paste paradigm. In large corporations, procurement officers spend a relatively large amount of time on minor variants of essentially routine purchases. Some report that up to 80% of orders are repeats of earlier ones. Ec^x exploits this to free procurement officers to focus their attention on the non-routine purchases where their skills produce more added value

- User involvement in selecting what is required and then in being able to monitor what then happens brings a number of advantages. Firstly, it is more likely that the user will receive what is needed for his job and the user feels more responsibility for the equipment selection. Secondly, the user need not constantly distract the procurement department with telephone calls to enquire about what has happened to his order, leaving procurement officers to get on with more valuable work

- Ec^x removes purchasing leakage by reducing the risk of the frustrated user visiting his local PC store to buy a forgotten extra at an inflated price and then claiming it on expenses or out of petty cash. A user can see everything that is available to him at preferential prices through the customer's tailored catalogue and has less incentive to go it alone against corporate policy. This is particularly the case when the order cycle time has been sharply reduced to the degree described earlier.

Most corporate procurement managers confirm and prize gains of this softer sort from Ec^x. The outcome for them is a combination of fewer dissatisfied users needing to be placated, more hours of procurement officer time on issues which genuinely add value and control over a greater proportion of the corporate IT spend.

The Ec^x Implementation

On the customer side of the Internet connection, Ec^x typically uses a standard Web browser on the user's PC and the Order Manager custom application on the procurement officer's PC. However, a user may use the Order Manager interface and the procurement officer may use the browser one, depending on how exactly the process and roles on the customer side are organised. In some instances, browser-based use of Server Order Manager blurs this distinction .

Both the browser and Order Manager communicate with Multivendor Computing's systems across the Internet. The exact route traversed through routers, proxies and firewalls is individually constructed by and for each customer and is not of consequence. However, in all cases, security and privacy are important considerations and the Ec^x system routinely uses SSL encryption to this end.

On the Multivendor Computing side of the Internet connection, all communications go through the Ec^x Web server. This presents the HTML pages and maintains session control. It also holds identity, status and preference information for those using Server Order Manager. Behind this sits the Web buffer server which is responsible for switching traffic to and from the appropriate servers and services in Multivendor Computing's operational systems.

The Web buffer server connects in particular to the:

- Catalogue Services which contain the customer information such as identities, passwords and discount rates. These services also generate the catalogues and notify current prices and availabilities to the Ec^x Web server as they change in real time. This information allows the Web buffer server to build appropriately tailored responses to incoming requests

- Product Database which contains all product line information including descriptions, part numbers, delivery lead times, graphics and brochure information. The Catalogue Services create a view of this information to form an individual customer's catalogue

- Legacy Systems which provide conventional functions such as inventory control, ledgers and order processing. These legacy systems in turn connect to some of Multivendor Computing's external suppliers through the

EDI Gateway. This gateway may also optionally play a part in communicating with Multivendor Computing's customers for ordering, delivery notification and invoicing for those customers who wish to operate in that mode.

Through the co-operation of these system components, Ec^x provides its full range of functionality and ensures secure protection against outside interference from the Internet.

Lessons

Over the course of its existence as a development project and then as an operational system, Ec^x has been a learning experience for all involved. Arguably, what one sees today bears little resemblance to the original inspiration, but that is common in any pioneering venture as Ec^x most certainly was. A number of lessons stand out.

Ec^x demonstrated quickly and starkly that merchant systems suited to consumer purchasing are lacking in the corporate purchasing environment. Consumers are familiar with adopting vendors' selling processes and information requirements on a purchase by purchase basis. A corporation, particularly a large one, imposes its buying process on each transaction and this makes it quite wrong to approach it as if it were an individual consumer.

This realisation resulted in the two faces of Ec^x. For the user (see section on Customer Roles), the browsing, searching and selecting design to create an order are similar to those found on consumer merchant sites because the user is often both non-expert and unfamiliar with the site. He operates very much like a consumer. For the procurement officer, a completely different design applies which appeals to speed, efficiency and control. This is not the traditional consumer catalogue with the "basket" paradigm. Order Manager is the Ec^x way of presenting an alternative, much more appropriate interface in these circumstances.

The corollary to this is that Ec^x must adapt to fit the corporate customer's procurement processes. It is only logical to expect a procurement department dealing with all classes of products to have no interest in a special process for IT, or even PCs as part of IT. The aim of Ec^x is to fit into the pre-existing corporate purchasing system. This intention had significant cost implications for the take-up of Ec^x by its early customers because each one appeared to be very different. However, as the number of customers grew, similar processes started to appear and the incremental cost of adding each further customer declined. Today, Ec^x has the ability to fit into all the common arrangements for procurement and the implementation team's accumulated expertise and interfacing code is now one of the most valuable assets of Ec^x.

The order status monitoring facility, although obvious in concept, has repeatedly proved to be one of the most applauded facilities offered by Ec^x. None of those who originally designed it into the system correctly predicted how much unproductive time procurement officers spend on updating users on progress. Without Ec^x-like monitoring, every user enquiry tends to result in another telephone call to the supplier followed by a call back to the user. Each enquiry might easily consume thirty minutes of man-time plus the elapsed time taken to restart any interrupted tasks. With Ec^x, the user can refer to his own screen, without disrupting anyone else's chain of thought. The anecdotal evidence is that as a result procurement teams can handle a far heavier workload for the same employment level and their lives are generally more ordered and pleasant.

As with all automation, Ec^x tends to show the differences between how things are supposed to happen and how they actually do. The early days of Ec^x forced Multivendor Computing to clean its own supply processes as a necessary pre-condition for success. Customers have found similar experiences. Without doubt this experience is beneficial, even if it is often an unexpected cost.

The Future

Ec^x has an expanding future. Undoubtedly, those responsible for it will consider refurbishment beneath the covers and may re-implement various parts of it to maintain its performance under growing workloads, to adapt to changing fashions in visual style and to ensure that it remains maintainable. There is also the on-going programme of extending the variety of customer legacy systems with which it will successfully

operate. A good example of this is the growing list of ERP systems with which it interfaces. All this is about making Ec[x] more successful in its current role.

An interesting exploitation of Ec[x] lies in its ability to feed information to ICL support and help desks. As an added service, callers will shortly be able to enquire about the state of their orders via voice calls to ICL call centres. But more importantly, an agent receiving a call for after-sales support or service can view the list of components which the caller has recently acquired. This makes ICL appear more professional, removes potential confusion about equipment identity and shortens call durations.

Ec[x] now allows customers to make available to its staff products for which Multivendor Computing is not the source. This is known as the multi-supplier facility and it uses a Web server located on the customer side of the Internet. Using the established Ec[x] software functionality, the corporate customer is able to rationalise an increasing proportion of its internal supply onto one set of processes. This could be just the start of continual expansion towards Ec[x] becoming a "one-stop shop" for a very wide range of business products, whether supplied through Multivendor Computing or not.

Geographic expansion beyond the current ten European countries is also clearly possible as Ec[x] now functions with non-English language facilities and currencies other than sterling. The multi-language facility operates through tags linking to tables of stored language texts. This makes it extremely easy to add another language simply by adding an additional table. The cost is little more than that attributed to one-off translation of the texts.

Some have suggested that Ec[x] could become the heart of a consumer-directed sales operation. While theoretically attractive, selling to consumers raises issues which have little or nothing to do with Ec[x] as such, but contribute materially to success in that market. Hence Multivendor Computing would have to handle credit arrangements, home delivery and warranty returns — all requiring the development of skills which Multivendor Computing has so far not needed in dealing with corporate customers.

An obvious further dimension for expansion is to package the program code of Ec[x] and to offer it for sale to other operators like Multivendor Computing. Of course, there are severe limitations to this dimension because ICL has no desire to lose the advantageous differentiation which Ec[x] lends to Multivendor Computing in its markets. However, there is a class of suppliers whose performance ICL does want to enhance. These are customers who distribute onwards to their customers products obtained from Multivendor Computing. These intermediaries could find just the same advantages as Multivendor Computing has done in using Ec[x]. Indeed, Multivendor Computing could host such services to minimise the investment cost for the intermediaries. Active investigation into this expansion route is on-going.

Conclusions

Ec[x] has conclusively proved that it can bring substantial benefits to all involved. Ec[x] is an excellent example of the assertion that the connected business world is more efficient than the unconnected one. Ec[x] produces gains in both elapsed and consumed time, reduces cost for both buyer and seller and increases the satisfaction of the individuals involved. Moreover, these benefits are often substantial.

For those who want to copy Ec[x] in a similar or completely different field, the lessons are largely those of any pioneering venture. The big wins are rarely predicted — for example, progress monitoring being so valuable — and the big challenges are often quite prosaic in nature — for example, interfacing to such a wide variety of legacy systems. The message comes through loud and clear that it is innovative, resourceful, dedicated professionals who create the success rather than technologies, let alone specific technological products, when there is no established custom and practice on which to rely.

Ec[x] now embodies much of that successful custom and practice. The professionals who create and operate Ec[x] have become experts in the field. The time for green field development in corporate purchasing schemes like Ec[x] is now over. Anyone wishing to emulate the success of Ec[x] should consult the experts and not attempt to start again.

Appendix

Screen shots showing the *modus operandi* of Ec^x are shown below.

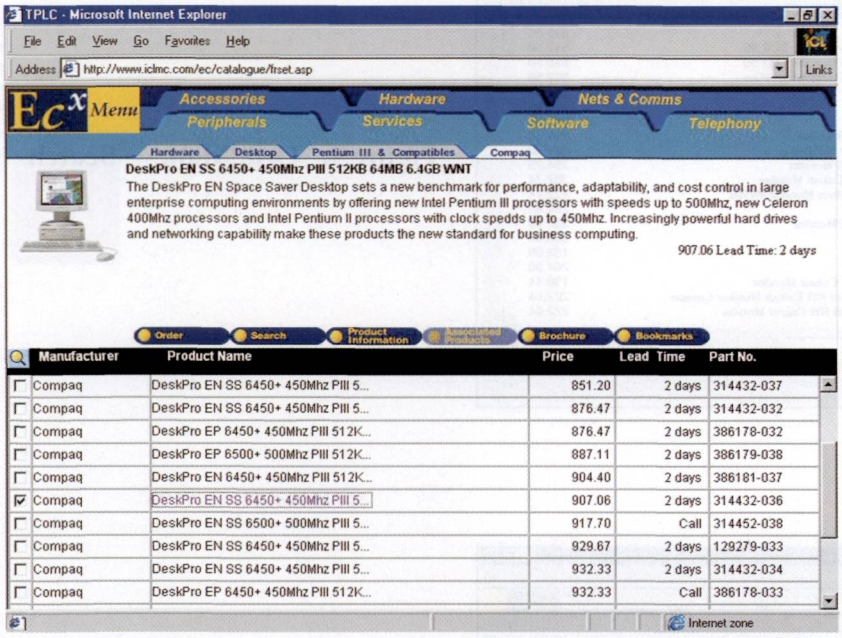


Figure 1: Catalogue

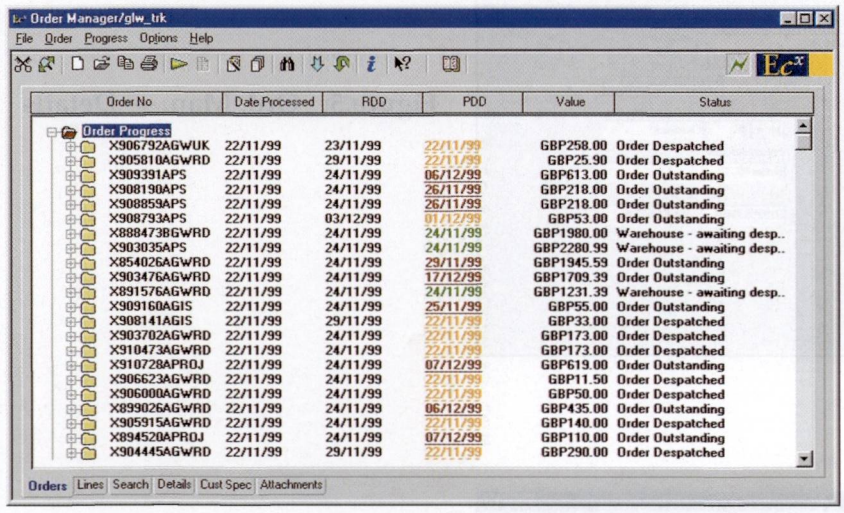


Figure 2: Ord. Man. — Orders

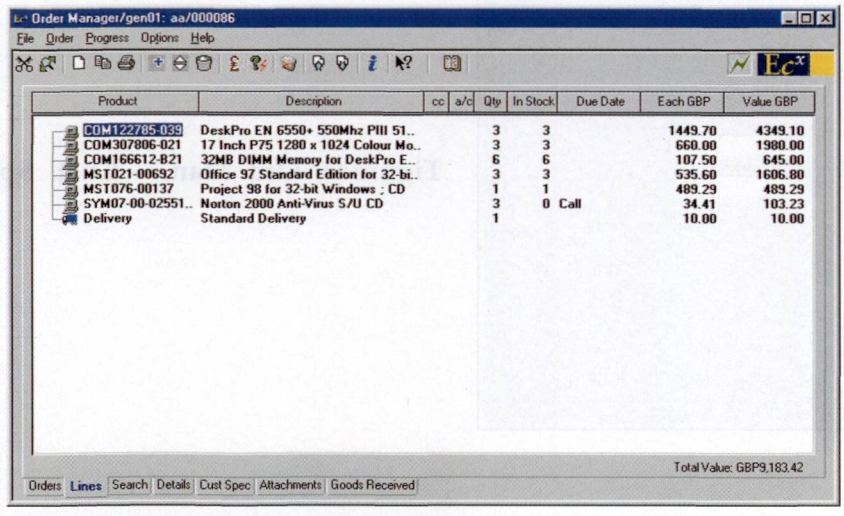


Figure 3: Ord. Man. — Lines

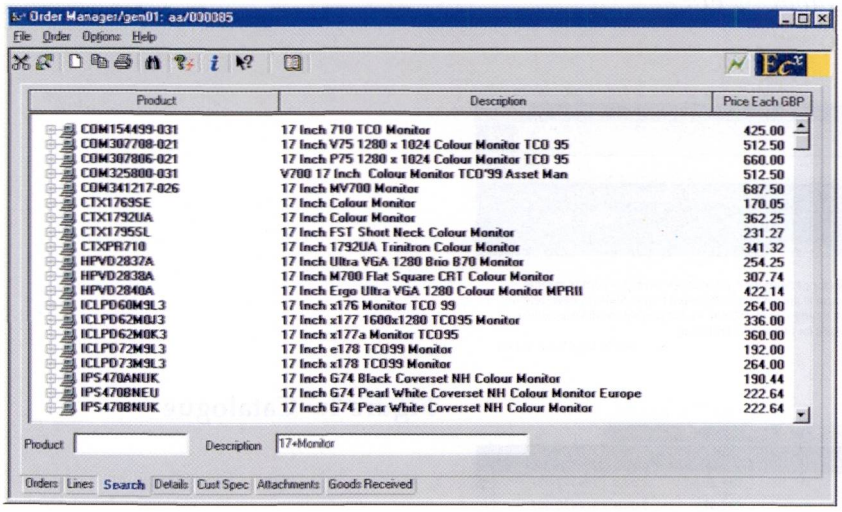


Figure 4: Ord. Man. — Search

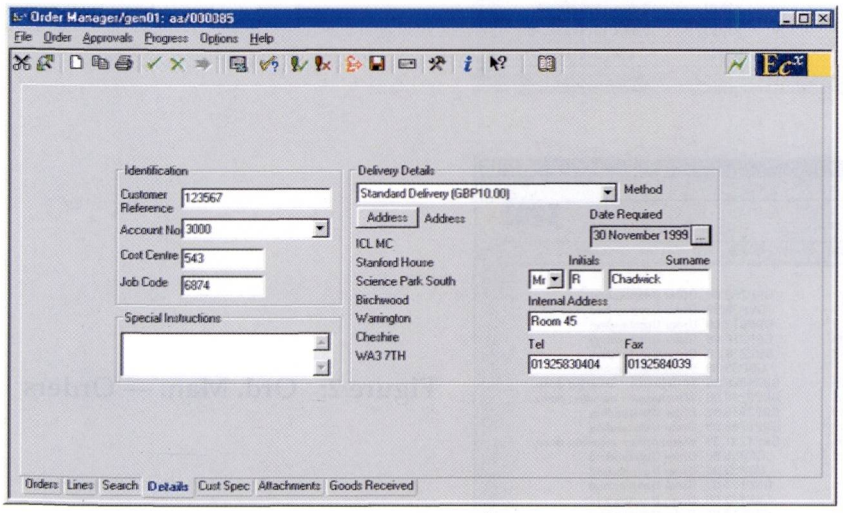


Figure 5: Ord. Man. — Details

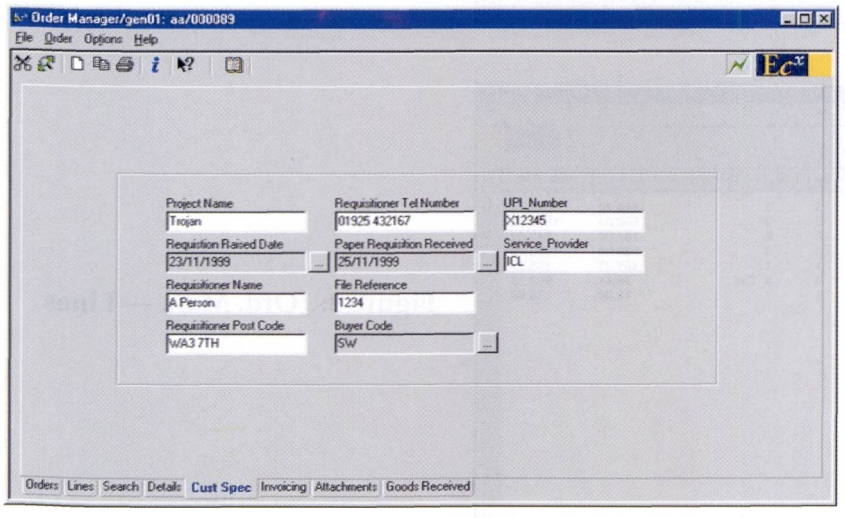


Figure 6: Ord. Man. — Cust. Spec.

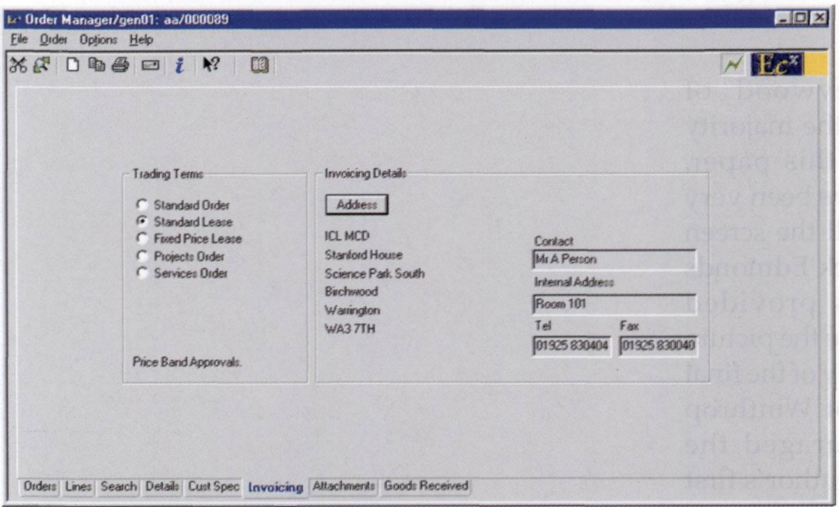


Figure 7: Ord. Man. — Invoicing

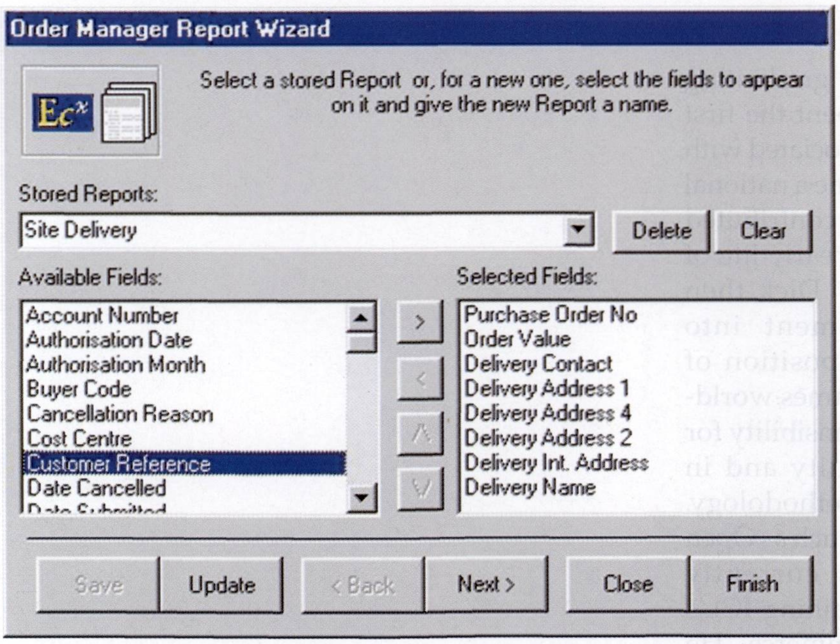


Figure 8: Report Wizard

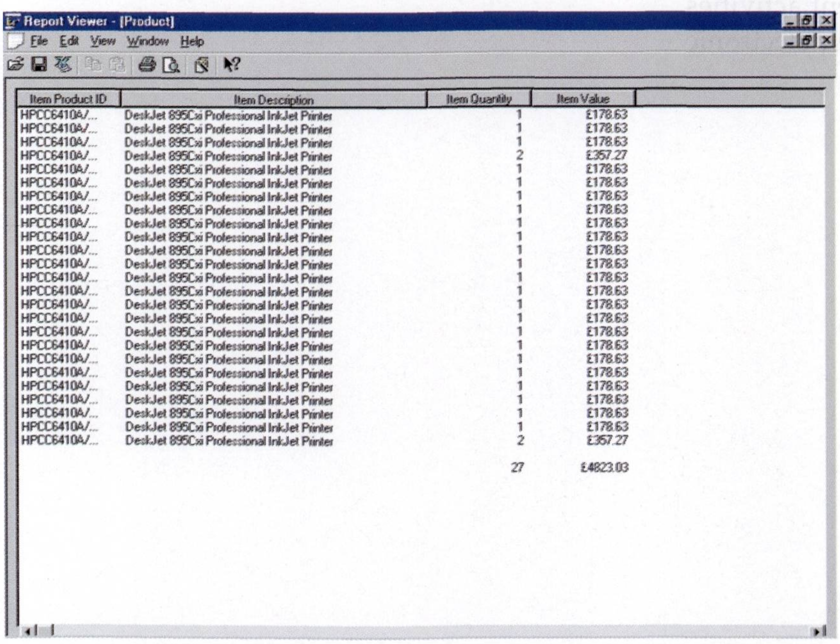


Figure 9: Report Viewer

Acknowledgements

Roy Chadwick and Steve Heywood of Multivendor Computing provided the majority of the information contained in this paper. Without them, this paper would have been very short indeed. Roy also provided all the screen shots which go with the paper. Nick Edmonds of Electronic Business Services provided additional technical detail to complete the picture and helped with ensuring the accuracy of the final text. Finally, Gabriel Engelhard of Winthrop Publications Limited first encouraged the creation of and then published the author's first paper on Ec[x] from which this paper derives.

Biography

Dick joined ICL immediately after graduating from Cambridge University. He spent the first half of his career in technical roles associated with sales and major customers. He became a national authority on operating systems and contributed managerially and technically to the early life of the ICL 2900 Series mainframes. Dick then moved through sales management into marketing. He rose first to the position of Marketing Manager for ICL Mainframes world-wide. He then took marketing responsibility for ICL's systems integration capability and in particular the OPENframework methodology. He also worked within related areas such as Open Systems and Intranets. He is currently responsible for creating and maintaining ICL's corporate strategy for electronic commerce. He represents ICL's electronic commerce viewpoint on a variety of trade and government activities as well as advising customers on their electronic commerce strategies.

EPIK: ENGINEERING PROCESS IMPROVEMENT AND KNOWLEDGE SHARING

Brian Chatters, John Hood, Nick Jefferson

ICL High Performance Systems, Manchester, UK.

Abstract

This paper describes a case study of how ICL High Performance Systems (HPS) has developed an "Engineering Process Improvement Framework" in order to deploy world class engineering practices. Projects define their processes in an *engineering definition* which is then assessed and calibrated against a model of best practice prior to implementation, leading to the identification of risks and actions to be managed by the project.

The core of the framework is an *engineering knowledge base*, implemented on the HPS Intranet, which contains a SPICE-conformant model of best practice (known as the *generic engineering definition*), options, tools, methods, and experience. This information is used by projects to create their *specific engineering definitions*, which document their *quality management systems*. The specific engineering definitions are held in the *engineering knowledge base* as objects for potential reuse by other projects.

The effectiveness of the framework is reviewed and measured against the business objectives of the project and lessons learnt are documented and published in the knowledge base.

Results from research activities and internal and external innovations are fed into the knowledge base to provide ongoing improvement to the definition of best practice.

Introduction

As part of our culture of continuous improvement, we recognised the importance of developing our core engineering competencies, methods and products in order to achieve industry best practice. A process improvement initiative was established within High Performance Systems (HPS) and a strategy developed to achieve this goal.

The initiative created a framework, known as EPIK (Engineering Process Improvement and Knowledge sharing) to meet the overall objective. The framework is an implementation of an experience factory [Basili, 1994].

Background

Starting Scenario

Historically, many of the large-scale enterprise systems and solutions have been developed in house by HPS. This approach has led to the establishment of effective, specialised processes, methods and tools, many of which have also been developed in house, to ensure that the systems and solutions meet the stringent quality requirements demanded by our customers. HPS was, and remains, responsible for ICL's large servers and the OpenVME operating system. More recently, third-party components have become more significant in the solutions offered by HPS. This has resulted in a clash of cultures and challenges to long held assumptions. Working with third parties can also introduce a number of unknowns into the development activities, increase risks and jeopardize delivered quality.

Further, organizational changes have led to a new style of working with a move away from the traditional functional departments such as marketing, development, testing, etc. to a project-based structure. Projects are charged with end-to-end responsibility, from the establishment of requirements through to customer delivery.

Projects operate autonomously and have total responsibility for delivering solutions against agreed budget, time scale, and quality criteria. This organizational approach ensures more flexibility in meeting new challenges and ensures that the responsibility for satisfying requirements is taken by the project team. However, the approach does have some potential risks.

- There is a possibility that each project will redefine the methods and processes to be deployed when planning its activities. Rather, projects need to adopt and tailor existing processes to meet their evolving requirements.
- The lack of fixed departmental structures can be a barrier to the sharing of experiences.
- It can be difficult for a project to give priority to improvement initiatives, given that there is a potential conflict of interest between delivering the product on time and within budget (business drivers) and getting the product right (customer driver).

Business Objectives

The changing nature of ICL's business (working with third party suppliers and collaborators through a project-based organization) necessitated processes, methods and tools becoming more flexible and ensuring a seamless integration of the interfaces, both product and process, with everyone involved in the end-to-end process.

HPS's primary objective with EPIK was to improve the predictability of costs and delivery dates for its systems and solutions. However, business pressures also demand continuous improvements to product quality, time to market, and productivity.

What is EPIK?

EPIK is a framework for making continuous improvements to the way in which ICL engineers systems and solutions to customers' requirements. In effect, it defines the *quality management system* [ISO, 1994b] used by engineers and managers when embarking on a project. Many implementations of process assessments are based on carrying out a health check against existing practices. The implementation of EPIK is based on the principle of prevention [Crosby, 1984] by carrying out assessments in a timely manner to avoid problems arising. Maximum benefit is achieved by exploiting reuse, best practice, assessment, and process tailoring.

Principles of EPIK

The implementation strategy for the EPIK framework is based on three principles:

Principle 1: *Project members identify with and accept responsibility for processes they deploy.*

To this end, an **Engineering Authority** role is defined. The purpose of the role is to provide leadership in establishing and implementing appropriate engineering processes and tools. The *engineering authority's* role is to co-ordinate the project's activities with the way team members perform their tasks.

Principle 2: *The processes deployed within a project need to be defined and assessed for their suitability in meeting the business objectives of the project.*

Typically, a project will require a technical specification of the product (system or solution) and a resource plan, in order to obtain funding. To reduce risk further in the project plan, the concept of an **Engineering Definition** has been introduced, which describes the processes (methodologies) that a project will use.

Principle 3: *Learning is at the core of an organization's ability to adapt to the rapidly changing environment of new ideas and innovations.*

Effective sharing of knowledge, experience and learning across the engineering community is provided by the **Engineering Knowledge Base**.

Requirements of EPIK

HPS has a long history of innovative development environments which help projects to deliver world class systems and solutions. Systems and tools such as CADES [McGuffin et al, 1980] and CHISLE [Jebson et al, 1993], backed up by sound engineering practices based on ISO quality system standards [ISO, 1994b], have played a major part in establishing best practice. The rapidly changing world of IT has highlighted the need continually to enhance and improve our

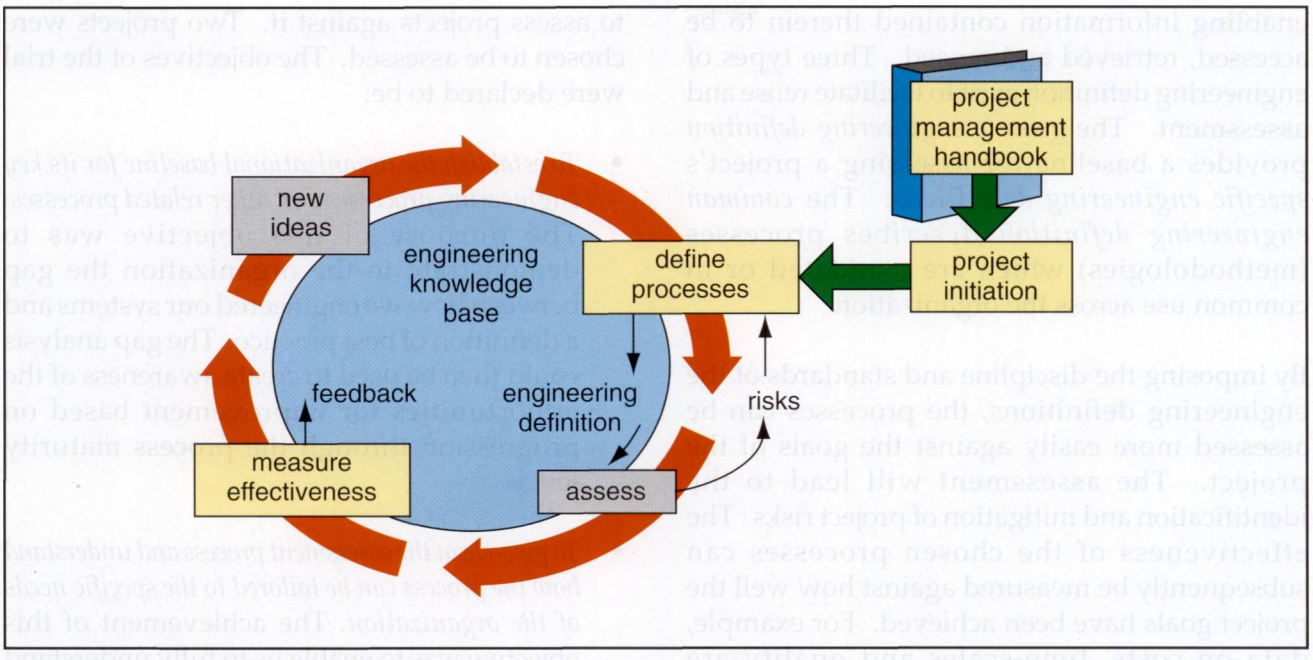


Figure 1: The EPIK framework

engineering methods. We recognise that there are no quick fixes or easy answers [Brooks, 1987]. Rather, a framework is required to enable the evolution of the engineering processes and methods in the rapidly changing environment in which we operate.

The medium-term requirements on the framework are that it should:

- promote the concepts of a "learning organization" in a systems engineering context
- establish a set of core engineering processes, essential for the development of any programme of work
- provide a method of evaluating the systems engineering processes and their deployment against the world's best practice
- establish mechanisms for defining, selecting and implementing improvements to processes and methods.

The framework has been constructed such that it can be extended to integrate the broader disciplines of engineering to meet the longer-term requirements:

- establish mechanisms for defining, selecting and implementing improvements to the tools and technologies
- to exploit research and innovations in the practice of systems engineering
- to develop an engineering information strategy, including mechanisms for the transfer to the engineering community of technological innovations in the ways that systems are developed.

Overview of EPIK

The *engineering knowledge base* is at the core of the EPIK framework; see Figure 1. Conceptually, this is the pool of all knowledge owned by the engineering community. The knowledge base is structured for learning, introducing new topics and supporting them with examples from project experiences. It is implemented as a web site on the HPS Intranet with a database of existing quality system documentation, best practice definitions and existing on-line control systems. It provides the framework in which best practice can be effectively deployed within projects. It is applicable to all engineering disciplines and provides generic processes based on world best practice, reflecting leading research ideas and internal innovations.

"Project initiation" requires processes to be defined, which describe how the project activities will be carried out. The processes are described in an engineering definition, which is published on the Intranet, thus giving it visibility, and

enabling information contained therein to be accessed, retrieved and reused. Three types of engineering definition exist to facilitate reuse and assessment. The *generic engineering definition* provides a baseline for assessing a project's *specific engineering definitions*. The *common engineering definition* describes processes (methodologies) which are mandated or in common use across the organization.

By imposing the discipline and standards of the engineering definitions, the processes can be assessed more easily against the goals of the project. The assessment will lead to the identification and mitigation of project risks. The effectiveness of the chosen processes can subsequently be measured against how well the project goals have been achieved. For example, data on costs, time-scales and quality are compared with original estimates or targets. Feedback and lessons learnt from the effectiveness review are published on the *engineering knowledge base* for consideration by other projects. Follow up actions may lead to improvements in the *common* or *generic engineering definitions*. The final operation within the framework is the ability to evolve best practice through the capture of new ideas. Such ideas may arise from research, through industrial or academic contacts, or be generated through internal innovation.

Making EPIK Happen

The implementation of EPIK was preceded by a feasibility study, the results of which led to the creation of the *generic engineering definition*, development of supporting processes and tools, and the establishment of awareness and training events.

Deployment of the framework was supported by a declared policy to underwrite senior management buy-in, and facilitated sessions were held to help the projects to establish their *specific engineering definitions*. This is now the norm.

Feasibility Study

The feasibility study was conducted by participation in the SPICE trial [Chatters, 1997a], SPICE is a widely used process improvement model. The trial enabled us to become familiar with the SPICE standard and to understand how to assess projects against it. Two projects were chosen to be assessed. The objectives of the trial were declared to be:

- *To establish the organizational baseline for its key engineering processes and other related processes.* The purpose of this objective was to demonstrate to the organization the gap between how we engineered our systems and a definition of best practice. The gap analysis could then be used to create awareness of the opportunities for improvement based on progression through the process maturity levels

- *To pipe-clean the assessment process and understand how the process can be tailored to the specific needs of the organization.* The achievement of this objective was to enable us to fully understand the SPICE model, the assessment process, the opportunities for customizing the model and process to the specific needs of HPS, and the costs involved in the implementation

- *To identify opportunities for improvement across the organization and within the two pilot projects, which will deliver business benefit.* This objective was to allow the demonstration of real improvements to the performance of the projects. Specifically, it needed to demonstrate added value to alternative assessment methods such as ISO/TickIT audits [ISO, 1990] and assessment against the EFQM Business Model [EFQM, 1998].

The trial used an assessment process, toolkit, and a SPICE-conformant process model, provided by an external consultancy. The trial also allowed us to assess the external process model as a suitable *generic engineering definition*. The trial was effective in establishing the HPS baseline and provided a firm foundation for the development of EPIK. The objectives of the trial were completely satisfied and the main findings were:

- The mapping between the external process model and the processes operated by the pilot projects was complex due to the devolved management responsibilities and the broader scope of activities deployed by HPS. Consequently, we decided to develop our own SPICE-conformant process model, tailored to our specific needs — the *generic engineering definition*.

- Measuring processes against maturity levels (showing how fully developed the processes are) provided an effective way of highlighting the key areas that needed attention. Relating the measured levels to targets for individual processes within the business context of the projects being assessed quantified the gap and allowed the issues to be prioritized. However, this level of flexibility in target setting had limited benefit to the organization. So we decided to use a simpler approach. A universal target will be set for all processes, and assessment PRIOR to deployment of planned processes will highlight shortfalls against this target. Such shortfalls will relate to risks, which can be addressed by process improvement or they can be tolerated (with appropriate contingencies built into the project plans), depending upon the business objectives.

Awareness and Training Events

Awareness and training are vital to the success of any programme of change. For EPIK, a number of events were organized:

- A half-day awareness event for all engineers and project managers, to enable them to understand capability maturity levels, the business benefits of improving process capability, the role of the *engineering authority* and *engineering definition*, and how to share experience and learning via the *engineering knowledge base*
- A one-day training event for *engineering authorities* to ensure that the organization is adequately prepared to implement the EPIK policy, to enable them to generate a project's *specific engineering definition* from the HPS *common engineering definition*, and to enable them to assess an engineering definition against the *generic engineering definition*
- Facilitated workshops with project managers and *engineering authorities* to help them initiate the process of creating engineering definitions. The key outputs from the workshop are the *responsibility matrix* and the *worry list* (see Figure 5 in a later section).

Management Commitment

The EPIK programme is endorsed and overseen by a senior management review body, chaired by the division's Technical Director. The agreement and publication of a policy statement, authorized by the Technical Director demonstrates further management commitment:

> *"All projects will use the HPS Engineering Process Improvement and Knowledge sharing framework (EPIK) to define and implement world-class engineering processes, procedures, standards, tools and methods, which are tailored to meet the business goals of the project."*

The policy declaration allowed the introduction of a plan to phase in engineering definitions for all current and future projects.

Linking EPIK to the Product Life Cycle

The establishment, assessment, and maintenance of a project's *specific engineering definition* is intimately linked to the product life cycle's quality checkpoints; see Figure 2. Consequently, our standard project management control and review processes give assurance that EPIK is being implemented in line with the declared policy.

Checkpoint 1:
Business case and Investment review.

Checkpoint 2:
Baseline review (following project initiation). The review includes a check that the engineering definition has been established and assessed against the model of best practice. Process maturity levels have been determined and risks identified. This review enables preventive action to be taken by introducing process improvements prior to implementation.

Checkpoints 3/4:
Pre-announcement review and release assessment review. The reviews include a check that the projects are conforming to the agreed engineering definition. The assessments give the project an opportunity to make further improvements if the chosen processes fail to meet the business needs.

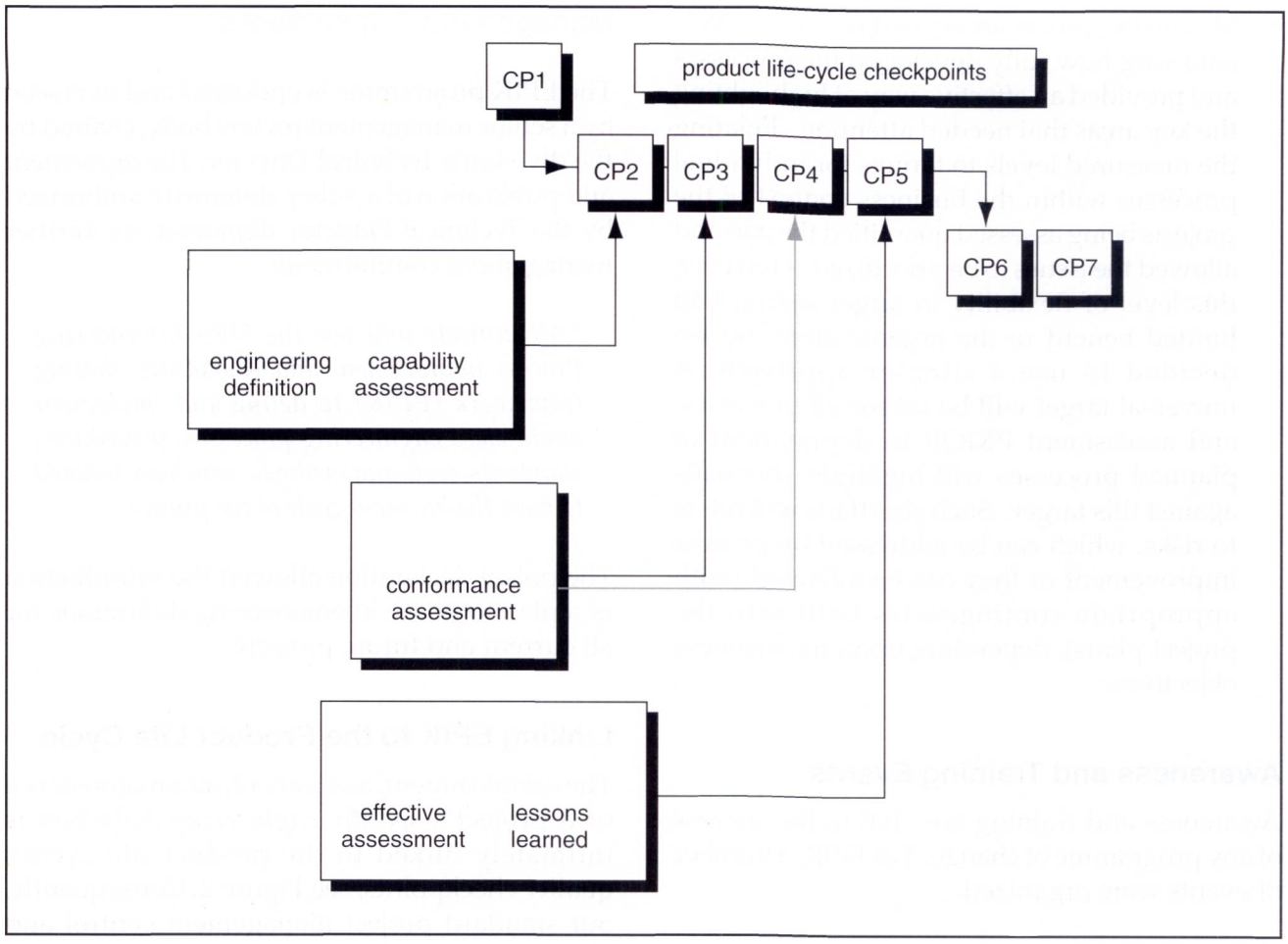


Figure 2: Product Life Cycle Checkpoints

Checkpoint 5:
Early life review. The review includes an assessment of the effectiveness of the project's processes against the business goals. Lessons learnt (good and bad) are recorded.

Checkpoints 6/7:
Mid life and product withdrawal reviews.

Development of the Engineering Definitions

We decided to develop our own process model — the *generic engineering definition* — following our experience of using an external model for the SPICE trial. It was intended originally to base the *specific engineering definitions* on our generic model but early implementations highlighted the need for a more effective means of documenting processes in a way that could easily capture best practice and reuse, or adapt, the ways of other projects. Consequently, we developed a standard for the layout of the definitions and we produced the *common engineering definition*, which defines default processes that are applicable to all projects. Specifically, the *common engineering definition* defines the mandated organizational processes. The standard layout also facilitates the reuse of processes described in *specific engineering definitions*. A schedule for the production of *specific engineering definitions* for current projects was agreed and implemented.

Relationships between the Engineering Definitions and other models

We had a legacy of processes based on our implementation of ISO 9000-comformant quality systems, quality improvement using Crosby's 14-step process [Crosby, 1979], and business development using the European Business Model [EFQM, 1998]. Consequently, it was important to evolve from the current systems and to integrate best practice from these systems into our model. Specifically, we:

- Structured our organizational processes round the enablers of the business model; see

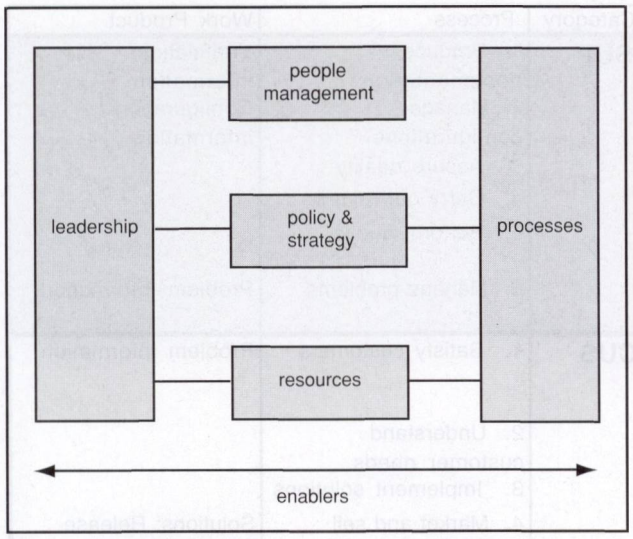


Figure 3: Enablers of the Business Model

Figure 3. The "Processes" enabler maps on to two processes, "Define Processes" and "Improve Processes". There is a one-to-one of the other enablers to processes within our model

- Extended the scope of our model to cover all aspects of systems and services development, including component acquisition, firmware, hardware, and software development, integration, and service delivery

- Ensured that the requirements of ISO 9001 and TickIT were captured in the key tasks. A mapping was also done against the ISO standard for software life cycle processes [ISO, 1994a]
- Based its structure on the SPICE reference model.

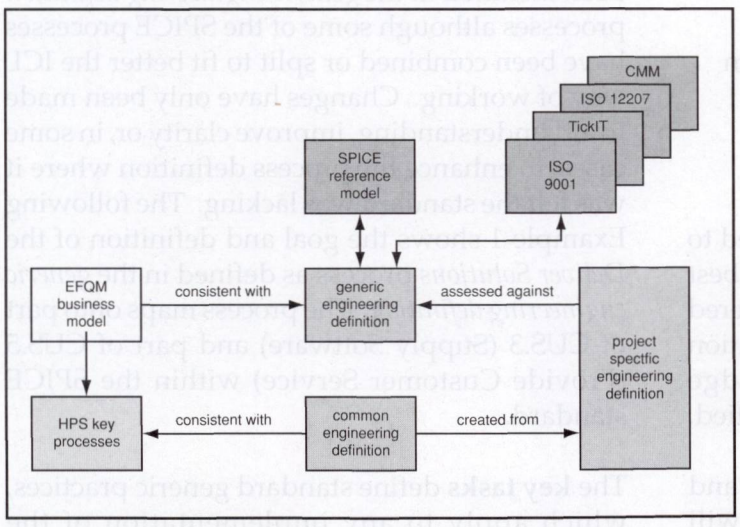


Figure 4: Relationships between the Process Models

Figure 4 illustrates the relationships between the various models. The *common engineering definition* describes the organizational-wide processes and includes references to the division's key business processes, which have been developed in conjunction with self-assessment against the EFQM business model.

Development of the Generic Engineering Definition

Structure of the Generic Engineering Definition

The *generic engineering definition* consists of twenty five key processes, organized into the following five categories:

- **Organizational (ORG):** processes required to establish the right environment for the effective operation of processes

- **Management (MAN):** processes required to manage projects and processes

- **Engineering (ENG):** processes which contribute to the production of the key outputs from the project (typically, these outputs are solutions to meet customer requirements and may be components, systems or services)

- **Support (SUP):** processes, which define common generic practices, which can apply to the operation of any process. These common practices may be implemented by service departments or by the establishment of standards and interfaces, which ensures consistency across the operation of processes

- **Customer (CUS):** processes, which interact with the external customers.

Interaction between these processes is represented by the flow of work products, where a work product is any input required by the process or output produced by the process. There are eighteen generic work products in the *generic engineering definition*, which have been formulated from the one hundred and nine work products in the SPICE model.

Category	Process	Work Product	Category	Process	Work Product
ORG	1. Provide leadership	Culture	**SUP**	1. Produce documentation	Verification information
	2. Establish Policies and Strategies	Policies and Strategies, Business plans		2. Manage configurations	Configuration information
	3. Manage people	Skills base, Culture		3. Assure quality	
	4. Provide resources	Project infrastructure		4. Carry out reviews	
	5. Define processes	Process definitions and standards		5. Perform audits	
	6. Improve processes	Process definitions and standards		6. Manage problems	Problem information
MAN	1. Manage projects and processes	Project/process plans, Engineering definitions; Verification information	**CUS**	1. Satisfy customers	Problem information
	2. Manage suppliers	Components; Verification information		2. Understand customer needs	
				3. Implement solutions	
ENG	1. Analyze and define requirements	Technical specifications, Verification information		4. Market and sell solutions	Solutions: Release infrastructure
	2. Design components	Designs: Verification information			
	3. Implement and test components	Components: Verification information			
	4. Integrate solutions	Solutions: Verification information			
	5. Validate solutions	Solutions: Verification information			
	6. Deliver solutions	Solutions: Release infrastructure, Verification information			
	7. Maintain solutions	Solutions			

Table 1: Processes and work products within the generic engineering definition

Table 1 above shows the twenty five processes and their related output work products.

Process Descriptions

Each process within the *generic engineering definition* is described by:

- a goal
- a definition and the scope of its operation
- key tasks
- input and output work products
- its relationships with other processes.

The **goal** identifies the objectives, which need to be satisfied by the process in order to achieve best practice. The objectives need to be considered within the business context of an organization and the focus of an assessment is to judge whether such goals have been or will be satisfied.

The **definition** describes a generic process and its scope which, if effectively deployed, will satisfy the defined goal.

In order to maintain a close relationship with the SPICE standard, the purpose and description of every process in the standard has been mapped directly on to the goal and the definition of the equivalent process in the *generic engineering definition*. Care has been taken to ensure that all elements of the SPICE standard descriptions have been included in the *generic engineering definition* processes although some of the SPICE processes have been combined or split to fit better the ICL way of working. Changes have only been made to aid understanding, improve clarity or, in some cases, to enhance the process definition where it was felt the standard was lacking. The following Example 1 shows the goal and definition of the *Deliver Solutions* process as defined in the *generic engineering definition*. The process maps onto part of CUS.3 (Supply Software) and part of CUS.5 (Provide Customer Service) within the SPICE standard.

The **key tasks** define standard generic practices, which apply to any implementation of the process. Each process has three or four key tasks,

Goal

The purpose of the Deliver Solutions process is to ensure that quality solutions are capable of being delivered and installed by the customer or service provider to meet customer requirements.

Definition

The Deliver Solutions process confirms that all the development activities are complete. The process:

- packages the solution
- defines delivery and installation instructions
- makes the solution available for marketing, manufacturing anExample

Example 1: Goal and definition of the *Deliver Solutions* process as defined by the *generic engineering definition*

which need to be implemented to satisfy the goals of the process. Completion criteria are defined for the key tasks, which are used to judge whether the goals of the process are satisfied within the context of the business objectives of the project. The following Example 2 shows the key tasks for the *Deliver Solutions* process as defined in the *generic engineering definition*.

The base practices are largely based on the base practices in the SPICE model, but when formulating the checklists, consideration was also given to the ISO standards [ISO, 1994b], [ISO, 1994a]. This approach has identified areas of the SPICE standard, which were thought to be weak and has given rise to enhancement of the *generic engineering definition* base practices and, in some

Key Tasks

Reference	*Tasks*	*Completion Criteria*
ENG.6.1	Check all development activities are complete.	Project Manager (or independent agent) confirms completeness of development activities.
ENG.6.2	Check release infrastructure is in place.	All support agents and services are prepare (includes marketing, sales, manufacturing, serv providers, help desks, as appropriate).
ENG.6.3	Review the solution with all stakeholders (method may be by inspection, walk-through, formal meeting or other appropriate means – may be a local process or covered by SUP.4).	Review(s) held with stakeholders; any actions arising are progressed to completion.
ENG.6.4	Authorize the solution for release.	Release approved by appropriate level of authorit (taking legal and contractual considerations int account); authorization recorded.

Example 2: Key Tasks for the *Deliver Solutions* process as defined by the *generic engineering definition*

To aid understanding and interpretation of the *generic engineering definition*, each process is supported by a checklist of **base practices**, structured under the key task headings for the process. They are in the form of questions which, if answered positively (or are not applicable), ensure all the right tasks are carried out within a project. A checklist of **management practices**, applicable to all processes, allows an assessment of how well the tasks are carried out and enables a maturity level to be assigned to an implementation of a process.

cases, additional base practices. The following Example 3 shows the base practices for one of the key tasks of the *Deliver Solutions* process as defined in the *generic engineering definition*.

The checklists are advisory only and are used as an aide-mémoire when generating or assessing specific engineering definitions.

Generic **input and output work products** are defined for each process and comments are added to clarify the work products within the context

Base Practices

Reference	*Practice*	*Spice (ISO 12207) Ref.*
ENG.6.2	Check that release infrastructure is in place	
ENG.6.2.1	Have interfaces to independent agents and subcontractors been determined? Include service and manufacturing organization.	CUS 3.3
ENG.6.2.2	Has product support been established? Consider. • Definition and establishment of a service to enable customers to raise problems and questions relating to the use of the system or product and to enable them to be resolved • Implementation of the service to enable customers to raise problems and questions relating to the use of the specific system or product being released and to enable the problems to be resolved. • Provision of self-diagnosis and help facilities to improve the effectiveness of the support mechanisms.	CUS 5.2 (5.2.7.2)

Example 3: Base Practices for the ENG.6.2 key task of the *Deliver Solutions* process as defined in the *generic engineering definition*

of the process, where relevant. Example 4 shows the input and output work products for the Deliver Solutions process as defined in the *generic engineering definition*.

definition will point to the supporting process definition, as it should always be the intention to provide generic supporting processes. This is not necessarily always the case, and supporting

Inputs	Outputs
• Project/process plans • Engineering definitions • Solutions	• Release infrastructure • Solutions • Process management information • Verification information • Feedback information

Example 4: Input and output work products for the *Deliver Solutions* process as defined in the *generic engineering definition*

Each generic work product is also defined by a purpose, description and list of specific work products, which would fall into this generic work product category. Additional information is also given to identify processes that use or output the generic work product. Example 5 shows the definition of the *Release Infrastructure* work product that is output from the *Deliver Solutions* process. The numbers in parentheses identify the corresponding work products within the SPICE standard.

The **supporting processes**, which are relevant to the generic process, are listed in the key tasks for that process. As can be seen from Example 6, the supporting processes specific to the *Deliver Solutions* process defined in the *generic engineering*

processes may be specifically defined in the context of a particular process.

Creating an Engineering Definition

The Common Engineering Definition

The *generic engineering definition* is a model of best practice for all the processes that may be operated by a project in the organization. The Organization (ORG) processes defined in the model are, by and large, operated by the organization and are equally applicable to all projects. Recruitment, supply of IT and provision of the office environment are but three examples of such processes. These processes are therefore defined in an organizational level document known as

Purpose

The *Release Infrastructure* contains all the deliverables that are needed to enable the marketing, selling, manufacturing, installation, operation and maintenance of a system, product or service.

Description

The *Release Infrastructure* includes marketing collateral, manufacturing information, installation guides, training, customer communication channels, service (help) desks, repair mechanisms, et cetera.

Contents Checklist

- Release information (release notes) (71)
- Installation guide (75)
- Delivery instructions (76)
- Handling and storage guide (80)
- Customer support procedures (82)

Affected Processes

Output From:	*Input To:*
ENG.6 Deliver solutions	ENG.7 Maintain solutions CUS.4 Market and sell solutions
CUS.4 Market and sell solutions	CUS.3 Implement solutions

Example 5: Definition of the *Release Infrastructure* as defined in the *generic engineering definition*

Base Practices

Reference	Practice	Spice (ISO 12207) Ref.
SUP.1	Produce Documentation	See SUP.1
SUP.2	Manage Configurations	See SUP.2
SUP.3	Assure Quality	See SUP.3
SUP.4	Carry Out Reviews	See SUP.4
SUP.5	Perform Audits	See SUP.5
SUP.6	Manage Problems	See SUP.6

Example 6: Supporting processes specific to the *Deliver Solutions* process point to the supporting process definition

the *common engineering definition*. This may also be true for any of the Engineering, Management, Support and Customer processes defined in the *generic engineering definition*, where the process is common to all projects. The *common engineering definition* provides a vehicle for a central definition of any process that is applicable across the organization.

The *common engineering definition* is structured in the same manner as the *generic engineering definition*. It contains all twenty five of the generic processes, with the goal and definition of each process repeated verbatim. For each process, the key task definitions become process descriptions and the completion criteria is replaced by a description of the procedures and standards to be followed in executing the process. Where a detailed procedure is required, this may be a reference to further process documentation. In fact, HPS already had a set of "Key Processes", which are now referenced from the *common engineering definition*.

This is best illustrated using the example of the *Deliver Solutions* process used earlier. A procedure is in place for delivering all products to the market place, regardless of which project

Cross References

[1]	Release Assessment checklist	MSSG/QU/053
[2]	Release Assessment Signatories Form	MSSG/QU/054
[3]	Release Assessment Review Results Form	MSSG/QU/055

Key Tasks

Reference	*Process Description*	*Procedures and Standards*
ENG.6.1	Check all development activities are complete	Independent quality authority confirms completeness of development activities - see SUP.5
ENG.6.2	Check release infrastructure is in place	Prepare infrastructure using the release assessment checklist [1].
ENG.6.3	Review the solution with all stake holders	Verify readiness using the Assess Product for Release process: general reference [2: MSSG/QU/009]; [3]
ENG.6.4	Authorize the solution for release	Release approved by appropriate level of authority (taking legal and contractual considerations into account); authorisation recorded [2]

Example 7. *Deliver Solutions* process as defined in the *common engineering definition*

developed and delivered the product. Therefore, the *common engineering definition* defines the process as in Example 7 above.

If a process is not in place at the organizational level, then each project must define the procedures it will follow to satisfy the key tasks of the process. The *common engineering definition* cannot define the procedures in this case and the process definition will appear as in Example 8.

In this example, standards exist for tracking modifications but the mechanisms for ensuring that the modifications are incorporated into future releases are product-specific and, hence, locally defined.

Specific Engineering Definitions

The *common engineering definition* identifies all processes that must be defined specifically for each project, as no organizational procedure has been defined. The initiation stage of any project mandates that a *specific engineering definition* is produced to define the processes to be followed by the project.

The first step in producing a project's *specific engineering definition* is to identify, with the aid of the *generic engineering definition*, which of the 25 processes are within the scope of the project and assign responsibilities to project members. This is done by members of the project team and facilitated by the EPIK team, using a tool known

Key Tasks

Reference	*Process Description*	*Procedures and Standards*
ENG.7.3	Package known modifications into future versions of the solutions	For hardware, EM3 ensures modifications are included in future versions of the products [2]. MEAd provides traceability of all customer-initiated modifications (either enhancements or product faults). *Mechanisms for including known modifications into future versions of solutions are defined in the local engineering definitions.*

Example 8: The key task ENG7.3 of the *Maintain Solutions* process as defined in the *common engineering definition* requires additional tailoring by projects to produce a complete process definition

Project :	HPS Generic Processes																									
	1	2	3	4	5	6	MANagement																			
Engineering Authority							1	2	ENGineering																	
									1	2	3	4	5	6	7	SUPport							CUStomer			
																1	2	3	4	5	6	1	2	3	4
	Provide Leadership	Establish Policies and Strategies	Manage People	Provide Resources	Define Processes	Improve Processes	Manage Projects and Processes	Manage Suppliers	Analyse and Define Requirements	Design Components	Implement and Test Components	Integrate Solutions	Validate Solutions	Deliver Solutions	Maintain Solutions	Produce Documentation	Manage Configurations	Assure Quality	Carry Out Reviews	Perform Audits	Manage Problems	Satisfy Customers	Understand Customer Needs	Implement Solution	Market & Sell Solutions
Project Responsibilities Scope ->	X	X	D	T	D	X	T	T	T	T	T	T	T	D	D	T	T	D	D	D	T	T	T	T	T
Supporting Process Roles **Manager?**																									
Produce Documentation	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
Manage Configurations				d			d	d	d	d	d	d	d	d	d	d								d	
Assure Quality	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
Carry Out Reviews						d	d	d	d	d			d	d											
Perform Audits							d	d						d											
Manage Problems				d	d		d	d	d	d	d	d	d	d	d			d		d				d	
PM Handbook Roles																									
1 Business Manager		✓																							
2 Project Manager (incl sub-PM)	✓	✓	✓	✓			✓																		
3 Benefits Manager		✓												✓	✓										
4 Technical Authority		✓		✓			✓		✓																
5 Supply Chain Manager								✓						✓											
6 Manager of Sub Projects				✓			✓							✓											
7 Sub-Project Managers																									
7.1																									

Figure 5: Responsibility Matrix

as the *responsibility matrix* to record the results; see Figure 5. Any issues raised during this session are recorded and form a *worry list* for the project manager to act upon.

The matrix is populated by doing the following:

- Identify, with the aid of the *generic engineering definition*, the processes that are within the scope of the project. Processes are marked with T, D or X:

  T = within scope and requires further tailoring

  D = within scope and the default definition in the *common engineering definition* will be used

  X = the process and resultant work products are not relevant to the project.

- Identify any sub-projects and their managers/team-leaders, in addition to the default roles defined in the organization's Project Management Handbook. The Project Management Handbook is a defined standard within HPS, based on the PRINCE II methodology [Bradley, 1996]

- Assign each process within the scope to the projects/sub-project roles

- Check if ORG processes, as defined in the *common engineering definition*, meet the needs of the project. If not, identify any local tailoring required

- Determine the applicability of the supporting processes. Identify which supporting processes will be applicable across the project and assign responsibility for management of these processes.

The *responsibility matrix* forms the basis of a list of contents for the project's *specific engineering definition*. The *common engineering definition* can now be used as a template for the *specific engineering definition*, by:

- Deleting all processes deemed to be out of the project's scope

- Adding references to the *common engineering definition* for all processes that follow the standard process

- Adding specific tailoring to common processes where they deviate from the standard

- Defining processes specific to the project by populating the "Procedures and Standards" section of the Key Tasks table. Consider the reuse (adoption or adapting) of processes from existing *specific engineering definitions.*

The process of generating the *specific engineering definition,* causes project staff to consider applicability of all best-practice processes, but ensures that time is not wasted re-inventing processes that already exist. If all available common processes are followed, then the project need only focus on defining or reusing the processes specific to their own project. This enables maximum benefit to be gained from process reuse.

Development of the Engineering Knowledge Base

The *engineering knowledge base* was implemented initially as a Windows Help system [Chatters, 1997b], known as "The Systems Engineering Excellence Model". SEEM exploited hypertext technology to link existing quality system documentation, best practice definitions and on-line control systems. At that time, HPS was still organized around functional units.

With the improved capability of the Worldwide Web and the availability of effective development tools, SEEM was re-engineered as an intranet site and re-launched as the *engineering knowledge base* [Chatters, B., Jefferson, N., 1999]. It still retained, in the main, legacy documentation. It has been further enhanced and restructured to facilitate and promote the continual improvement of best practice engineering methods through knowledge sharing within the project-based structure of HPS. In addition to the engineering definitions the *engineering knowledge base* contains detailed project documentation and more general background information about relevant topics. The structure of the *engineering knowledge base* has been chosen to support organizational learning and in particular the reuse of experience.

Engineering Knowledge Base Structure

The *engineering knowledge base* Intranet site has been designed to encourage and support projects in adopting the practice of knowledge sharing. The overall structure of the site breaks down into three main groupings:

- Projects and processes — Engineering definitions and other project-specific documentation, common processes of interest to all projects

- Topic-based instructional material to introduce new concepts

- General background and further information to enable wider learning and incorporate best practice and innovation.

Figure 6 shows the general structure of the site. From the home page the reader can go directly to the various topic sections, link to indexes of current and historical projects and processes, or browse through collections of useful learning such as papers, URLs and contacts.

In the structure we reflect a two-level model of learning in which concrete and context-dependent experience is collected to support the learning process. The model hypothesizes that initial and basic knowledge can be gained from rule based methods and generalizations [Dreyfus & Dreyfus, 1986]. In order to achieve higher levels of knowledge (expertise) a large amount of context-dependent experience needs to be assimilated. Thus in the *engineering knowledge base* we present the general principles first of all in topic sections.

Topic sections have a simple linear structure where the user navigates from one document to the next in an almost "tutorial" style to learn about engineering definitions, life-cycle models, best practice, and engineering skills. Once the underlying ideas have been absorbed then we expect the user to delve deeper into the site, moving from the generic to the specific project documentation. The structure of the project-specific sections is hierarchical with several layers of detail from a top-level menu of the different projects, through the engineering definitions, to the standards, checklists and work instructions. In addition to topic and project-based material, the *engineering knowledge base* includes sources of further information produced both internally within HPS and externally. This further information takes the form of reports, journal articles and case studies held on the HPS Intranet server and referenced hyperlinks to ICL and external Web sites. In exploring the further information and project documents the user is

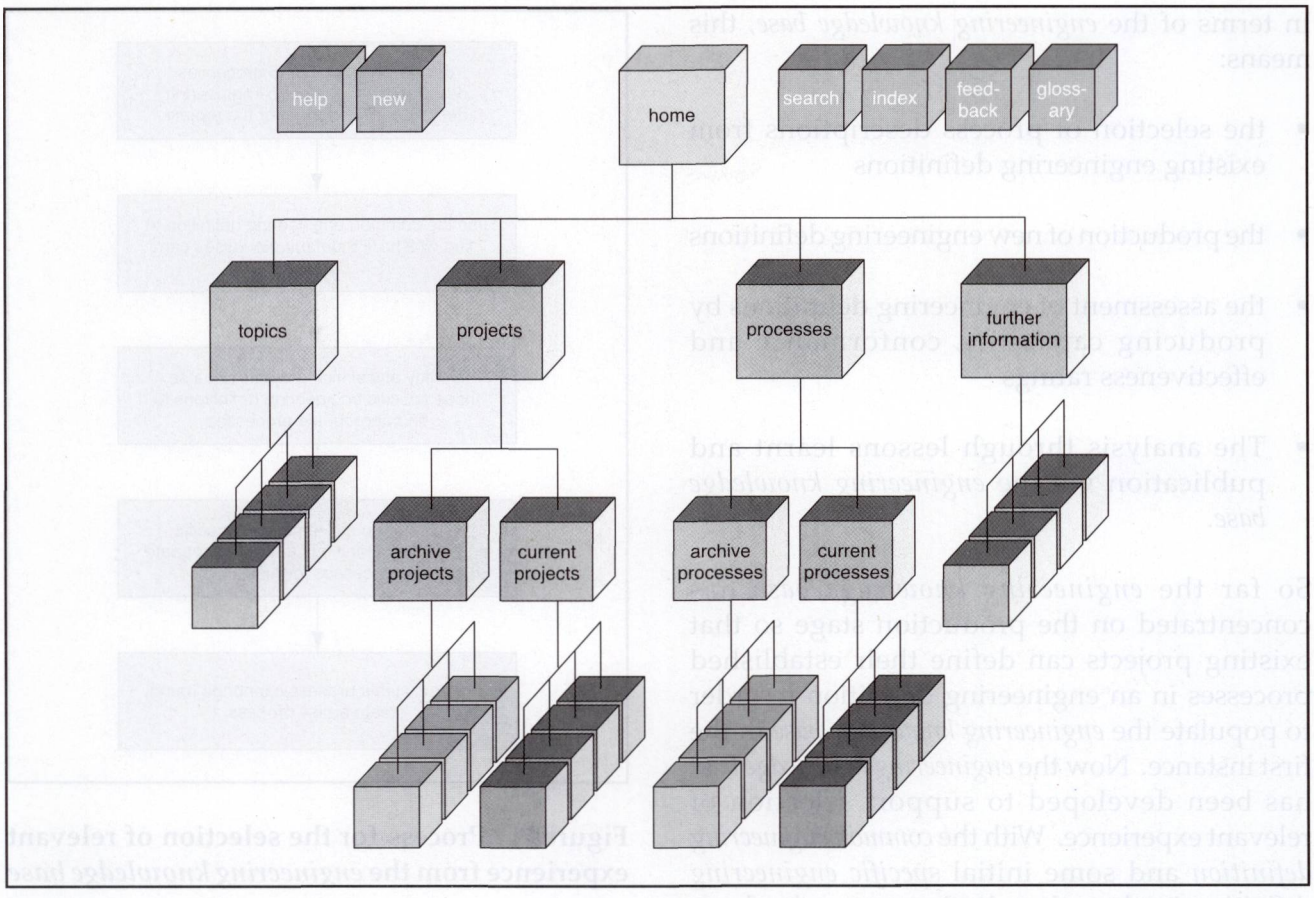


Figure 6: The General Structure of the HPS Engineering Knowledge Base Intranet Site

making choices about what is read and thus selectively acquires concrete examples of the previously introduced concepts, according to his or her own needs.

The project-specific content of the *engineering knowledge base* has been arranged in a project-centred structure. For each project a graphical index is maintained which relates the engineering definition and assessment documentation to the product lifecycle (as featured previously in Figure 2). As the project lifecycle progresses, the additional documentation is produced at each checkpoint and is registered in the *engineering knowledge base*.

This feature provides a central point for project staff to gain an overview of their project, relate their project to the context of its lifecycle and to other HPS projects, and to access standards, procedures, checklists etc. useful in their job tasks via a single point of access. But more important than this role, as a reference for project staff on existing developments, is the role of the *engineering knowledge base* as a central resource for reusable knowledge and experience.

Reusing Knowledge and Experience

In structuring the *engineering knowledge base* a model has been developed of how an organization reuses its accumulated knowledge and experience.

- The first stage in this process is to select those experiences which are relevant to a particular problem being faced.
- This selected knowledge is then acquired and applied to the new situation to produce something new.
- If the reuse process is to be complete and add something back to the pool of organizational knowledge then there must be some reflection on the success or otherwise of the project. Assessment produces a measure of success.
- Analysis then asks why this level of success was achieved. By capturing the analysis and feeding it back into the accumulated knowledge, so the sum total of the organizations learning is increased.

In terms of the *engineering knowledge base*, this means:

- the selection of process descriptions from existing engineering definitions
- the production of new engineering definitions
- the assessment of engineering definitions by producing capability, conformance and effectiveness ratings
- The analysis through lessons learnt and publication via the *engineering knowledge base*.

So far the *engineering knowledge base* has concentrated on the production stage so that existing projects can define their established processes in an engineering definition in order to populate the *engineering knowledge base* in the first instance. Now the *engineering knowledge base* has been developed to support selection of relevant experience. With the *common engineering definition* and some initial *specific engineering definitions* in place, the selection process has been described in the five stages shown in Figure 7.

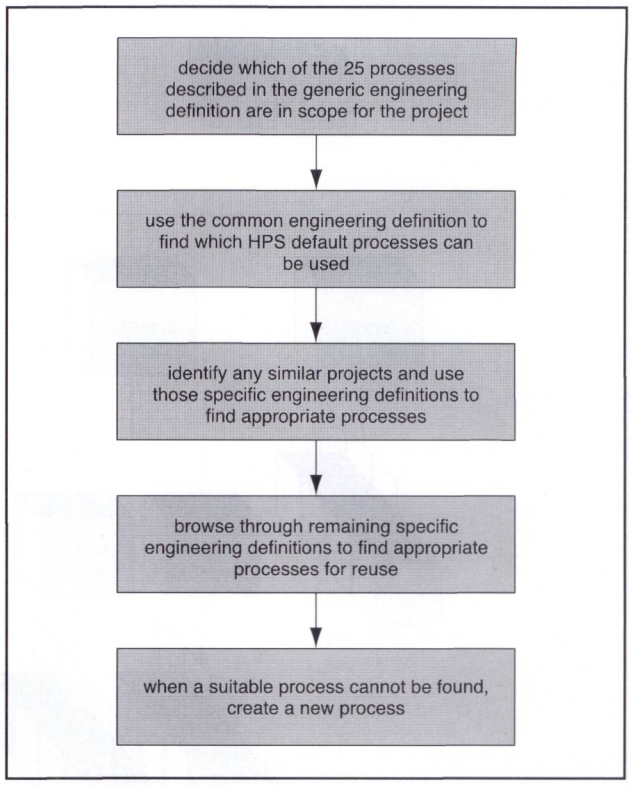


Figure 7: Process for the selection of relevant experience from the *engineering knowledge base*

The process descriptions in the *common engineering definition* are sufficient or at least largely sufficient for many of the organizational, management, support and customer processes. Where the default process is not sufficient then the *specific engineering definitions* should be consulted. HTML versions of the *specific engineering definitions* have been created and linked through navigational devices which enable users to focus on one of the generic processes and browse across various examples of that process; see Figure 8. Although this facility was requested by several of the engineering authorities responsible for producing engineering definitions, in actual fact the implemented facility has not been widely used. Instead, users have preferred to download previous engineering definitions in the original word-processed format. This may be due to various factors:

- Users prefer to print long and complex documents and read from paper rather than on-line
- The word-processed format remains the authoritative version, from which the HTML version is statically generated. Therefore, the word-processed versions are trusted as up-to-date
- The performance of the search mechanism is limited by current technology.

This method of downloading selected engineering definitions is sufficient while there is a manageable number of documents to read. As EPIK progresses and projects feedback more and more information into the *engineering knowledge base*, there will be a need for a more sophisticated tool to support users.

It is important that a process is carefully considered for appropriateness before it is reused. What has proved to be right for one project will not necessarily prove to be right for another project. When judging whether or not to reuse a particular process it is important to bear in mind the context in which that process was originally deployed. There are a number of criteria which *engineering authorities* should bear in mind.

- **Project attributes** such as size of team, number of components, and type of development (for example, software or hardware).

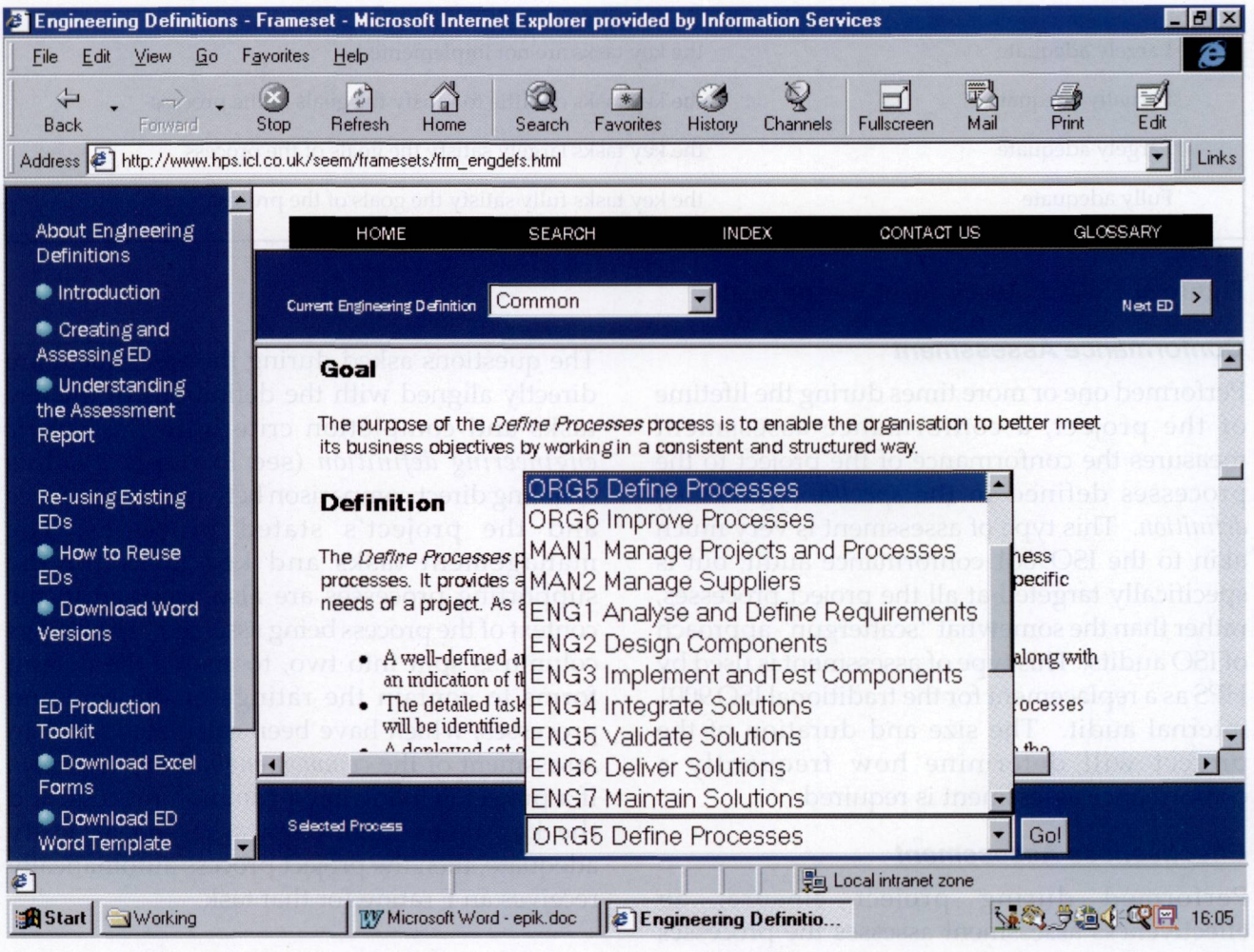


Figure 8: Browseable HTML engineering definitions in the *engineering knowledge base*

- **Project metrics** describe how successful a process was. Metrics are captured by the three assessments of the engineering definition according to capability, conformance and effectiveness and are included in the *engineering knowledge base*.
- **Lessons learned** should be considered in conjunction with the metrics as, with the benefit of hindsight, a process that did not work for one project could be successfully deployed for another project with appropriate modifications.
- **New innovations** with little in common with previous projects may be unable to find comparable *specific engineering definitions* to consult. In this case additional care should be taken before adopting previous processes.
- **Underlying technology can change**, rendering the candidate process inappropriate for reuse.

Engineering Definition Assessment

The effectiveness and deployment of the specific engineering definition are measured by assessments at defined stages of the project, determined by the product life cycle quality checkpoints. Personnel, independent of the project being assessed, undertake the assessment and the results are reported to the project review board. The assessment process also allows projects to carry out self-assessment to verify the adequacy of their *specific engineering definitions* prior to the formal assessments associated with the product life cycle process's quality checkpoints.

Types of Assessments

Three types of assessment are performed.

Capability Assessment

Performed during project initiation, a capability assessment assesses the processes defined in the *specific engineering definition* to verify the intent of the documented processes.

Largely adequate	the key tasks are not implemented
Partially adequate	the key tasks do little to satisfy the goals of the process
Largely adequate	the key tasks largely satisfy the goals of the process
Fully adequate	the key tasks fully satisfy the goals of the process

Figure 9: SPICE Assessment Rating Scale

Conformance Assessment

Performed one or more times during the lifetime of the project, a conformance assessment measures the conformance of the project to the processes defined in the *specific engineering definition*. This type of assessment is very much akin to the ISO9001 conformance audit, but is specifically targeted at all the project processes, rather than the somewhat "scattergun" approach of ISO audits. This type of assessment is used by HPS as a replacement for the traditional ISO 9001 internal audit. The size and duration of the project will determine how frequently a conformance assessment is required.

Effectiveness Assessment

Performed during project closure, an effectiveness assessment assesses the processes defined in the *specific engineering definition* to determine how effective the processes were in achieving the goals of the project. The results of these assessments are published in the *engineering knowledge base*, to enable future projects to gain from the experience of others.

Assessment Techniques

The techniques of assessment broadly follow those recommended by the SPICE standard. The key tasks for each process are rated on a 4-point scale; see Figure 9.

The scale is not linear and can be thought of as 0%, 30%, 70% & 100%. This is intended to force the assessor to decide if the key task requires major improvement, or the task has only minor deficiencies where improvements could be made.

Assessment Forms

The assessment process is aided by a set of forms, provided by the EPIK team. Example 9 shows part of the assessment form for the *Deliver Solutions* process, used earlier.

The questions asked during the assessment are directly aligned with the definitions of the key tasks and completion criteria in the *generic engineering definition* (see Example 2), thus enabling direct comparison between best practice and the project's stated processes. The management tasks and key tasks for the supporting processes are also assessed in the context of the process being assessed. The ratings column is split into two, to enable the default forms to contain the ratings of the common processes, which have been calculated from an assessment of the *common engineering definition*. If a project is following a common process, and the key tasks have been assessed as **F**ully adequate, then the project process automatically receives an **F** rating for that task.

An "observations form" is provided to record any observations made during the assessment. An observation is recorded against all key tasks that do not achieve an **F** rating.

Reporting results

All three types of assessment are seeking "opportunities for improvement". Issues are raised and recommendations are made and recorded during the assessment process for subsequent action by the project. On completion of an assessment, the results are tabulated and calculations are made to give an overall percentage achievement for each process against each of the maturity levels. Initially, we have set a target for all project processes to operate at maturity level three. Ultimately, our target is to achieve maturity level four.

The following Example 10 illustrates how the results are displayed. The radial chart presents a summary of the maturity of each process, highlighting those processes which have low maturity ratings. It also gives an indication of the shortfall at each level, highlighting those

Assessmment Process Engineering Definition	**ENG.6 Deliver Solutions** <ED Reference>	Date: Type:	
		Ratings	
Reference	*Tasks*	*Com*	*Spec*
Level 1	Process Performance		
ENG.6.1	Are all development activities complete?	F	
ENG.6.2	Is the release infrastructure in place? (includes marketing, sales, manufacturing, service providers, help desks, as appropriate)	F	
ENG.6.3	Have review(s) of the solution been held with stakeholders? ; Are actions arising progressed to completion?	F	
ENG.6.4	Is the release approved by appropriate level of authority (taking legal and contractual considerations into account)? ; Is the authorisation recorded?	F	
Level 2	Performance Management		
MAN.1.1	Does the project plan include estimates of costs and time-scales, and defined quality criteria for the completion of this process?	F	
MAN.1.3	Are regular status reports published showing progress against plans and achievement of quality criteria for this process?	F	
MAN.1.4	Are regular reviews held on status of this process? ; Is performance evaluated and actions identified to address any issues? ; Are actions progressed to completion?	F	
Level 2	Work Product Management		
SUP.1.1	Is documentation controlled within this process or defined in the scope of the common process?	F	
SUP.1.2	Are standards defined for all documentation in scope? ; Is conformance to standards demonstrated?	F	
Etc...			

Example 9: Part of the Assessment Form for the *Deliver Solutions* process

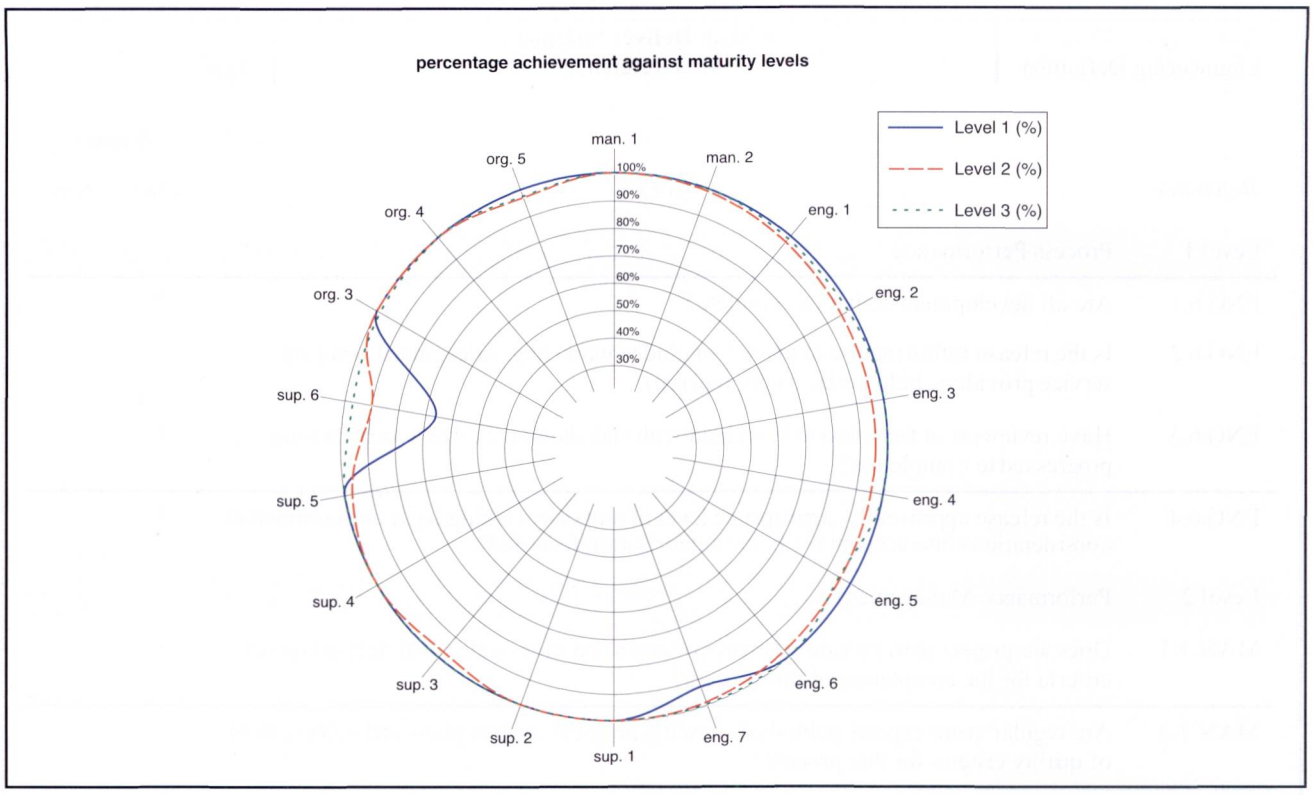
processes of particular concern. For example, it clearly illustrates that process SUP.6 falls far short of maturity level one.

Conclusions

There are no easy answers to the challenge of developing an organization's engineering capability. Implementation of the EPIK framework leads to continuous improvements in the predictability of project costs and delivery dates, reduced time to market, improved productivity, and the improved quality of delivered systems, solutions and services.

Engineering definitions have become the standard way to document how projects operate. To date, we have produced ten definitions and there has been an increasing improvement in their assessment ratings for the documented processes. The definitions are a combination of converting older quality system documentation and defining new processes. The *common engineering definition* has been adopted as the standard layout for engineering definitions and this adoption has greatly improved the ability to compare different definitions. It takes less than a day to carry out and report on a capability assessment.

The main benefit to date has been to reduce risks in achieving project deliverables within an agreed budget, on time, and with the required quality. This benefit has been realised by the early and timely identification of potential problem areas, enabling actions to be taken to mitigate risks. The types of problems that have been identified best



Example 10: Maturity rating summary expressed as a radial chart in the Assessment Report

illustrate this point. These problems are typical of the types of issues that do not normally get addressed early enough in the planning cycle.

- The need to revise the configuration management system to cater for a diverse set of components (previously, each component had its own configuration system but the controls for the integrated system were inadequate).
- The problem management system only dealt with customer problems. It needed extending to handle supplier problems and internal problems.
- The validation strategy was not sufficiently thought through early enough to enable delivery dates to be confidently underpinned.
- Consideration had not been given as to how the product would be maintained after its first release.
- No decision was made on how internal (non-deliverable) documents would be controlled and made accessible to project members.

There is a perception by project members that the framework has facilitated the transfer to the new mode of working but this perception is only backed up by anecdotal evidence. HPS's experiences with EPIK are now being exploited in the context of ICL-wide Knowledge Management.

Acknowledgements

Many of the ideas for the engineering improvement framework were influenced by numerous discussions with Peter Wharton, who was ICL's Chief Engineer. We would also like to thank Barry Hopewell and John Vernon of ICL HPS, and Peter Henderson of Southampton University for their continuing support for the programme and for their constructive suggestions and comments as it has proceeded.

Bibliography

BASILI, VICTOR R., *"Experience Factory,"* Marciniak, John J. (ed.) Encyclopedia of Software Engineering, New York: John Wiley & Sons Inc. pp. 469 - 476, 1994.

BROOKS, FREDERICK P., *"No Silver Bullet — Essence and Accidents of Software Engineering"*, Kugler, H-J (ed.) Computer, pp. 10–19, April 1987.

BRADLEY, K., *"Understanding PRINCE 2,"* Ken Bradley, Bournemouth, Dorset, UK,: SPOCE Project Management Limited, 1997.

CHATTERS, BRIAN, *"Process Assessment Summary Report,"* Internal Report. Available on-line at http://www.hps.icl.co.uk/seem/reports/assrpt.doc, 1997.

CHATTERS, BRIAN, *"The Systems Engineering Excellence Model,"* ICL Systems Journal, vol 11, 2, pp. 319 - 335, 1997.

CHATTERS, B., JEFFERSON, N., *"Engineering Knowledge Base,"* Intranet site. Available on-line at http://www.hps.icl.co.uk/seem/index.html, 1997.

CROSBY, PHILIP B., *"Quality Is Free,"* McGraw-Hill Book Company, New York, 1979.

CROSBY, PHILIP B., *"Quality Without Tears,"* New American Library, New York, 1984.

DREYFUS, H., DREYFUS, S., *"Mind Over Machine: The Power of Human Intuition and Expertise in the Area of the Computer,"* Free Press, New York, 1986.

EFQM, "The EFQM Model," http://www.efqm.org/model.htm, 1998.

ISO 1011 - 1, *"Guidelines for Quality Systems. Part 1: Auditing,"* 1990.

ISO 9000-3, *"Quality Management and Quality Assurance Standards - Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software,"* 1991.

ISO/IEC 12207, *"Software Life Cycle Processes,"* 1990.

ISO 9000, *"Model for Quality Assurance in Design, Development, Production, Installation, and Servicing,"* 1994.

ISO PDTR ISO 15504, *"Software Process Assessment, Part 2: A Reference Model For Processes And Process Capability, Version 2.0,"* 1996.

JEBSON, A., JONES, C., VOSPER, H., *"CHISLE: An Engineer's Tool for Hardware System Design,"* ICL Technical Journal, vol. 8, 3, pp. 506 - 519, 1993.

MCGUFFIN, et al., *"CADES — Software Engineering in Practice,"* ICL Technical Journal, vol. 2, 1, 1980.

PAULK, M. C. et al., *"Key Practices of the Capability Maturity Model, Version 1.1,"* Software Engineering Institute, CMU/SEI-93-TR-25, 1993.

Biographies

Brian Chatters

Brian Chatters is a Software and Systems Engineering consultant within ICL High Performance Systems. He is responsible for ensuring continuous development of the organization's software and systems engineering capability. He has worked in the software industry for over 30 years in various roles including programming, strategic design and project management. Over the last 12 years, he has focused on quality management and process improvement. He graduated from Bristol University with an honours degree in mathematics. He is a fellow of the Institution of Electrical Engineers, a Chartered Engineer and an active committee member of the BCS SPIN (UK) Special Interest Group. He is also a qualified ISO 9000 auditor and a qualified SPICE assessor.

John Hood

John Hood is a Quality and Process specialist within ICL High Performance Systems. He is responsible for ensuring project processes are in place to deliver products to time, cost and quality requirements. He has worked in the IT industry for 25 years in roles covering manufacturing, system support, hardware, firmware and software development, primarily in the Corporate Systems arena. During the last 6 years he has focused on process improvement across the development lifecycle and delivered quality of new products. He graduated from Leeds University with a degree in Computational Studies. He is a qualified ISO 9000 lead-auditor and a qualified SPICE assessor.

Nick Jefferson

Nick Jefferson is a Graduate Entrant within ICL High Performance Systems, currently working in the MSSG Data Access project. He has worked on implementing the EPIK framework in HPS by developing and maintaining the *engineering knowledge base* on the Intranet. He has also assisted projects in the production, publication and assessment of their engineering definitions. He graduated with honours in History from the University of Leicester in 1995. Whilst working at Glasgow University Library he became involved in the provision of information services over the Internet and decided to retrain in systems and software engineering. He graduated from UMIST with a MSc in Computation in 1998 and is a Microsoft Certified Systems Engineer (MCSE).

Previous Issues

Vol. 14 Iss. 1 – Autumn 1999

Interactive Digital Television
Internet Shopping Services
Online Loyalty in a U.S. Supermarket Environment
Using Scenarios to develop and implement strategy
The ICL Policy Scheduler for Windows
Performance—an Engineer's Guide

Vol. 13 Iss. 2 – Spring 1999

Project THOR
Monterey: A Web Content Production System
Cochise-2: An Integration Solution Kit for TPMS Applications
Engineering a Knowledge Utility: The National Grid for Learning
System Management in ICL's Millennium Programme
"Baby's" Legacy—The Early Manchester Mainframes

Vol. 13 Iss. 1 – Autumn 1998

Guest Editorial
An Architecture for Commercial On-line Internet Services
The Enterprise Datacentre—ICL's "Millennium" Programme
Trimetra DY and the Emulation of OpenVME on Intel Hardware
Trimetra UNS
Trimetra *Xtra*server
Millennium Data Access

Vol. 12 Iss. 2 – November 1997

Workflow—A Model for Integration
SuperVISE—System Specification and Design Methodology
Process Modelling using the World Wide Web—ProcessWise™ Communicator
Mobile Applications for Ubiquitous Environments
Middleware Support for Mobile Multimedia Applications
INDEPOL Client—A 'facelift' for mature software
Using the ECL^{i}PS^e Interval Domain Library in CAD

Vol. 12 Iss. 1 – May 1997

Java™—An overview
Mobile Agents—The new paradigm in computing
The SY Node Design
Discovering associations in retail transactions using Neural networks
Methods for Developing Manufacturing Systems Architectures
Demystifying Constraint Logic Programming
Constraint Logic Programming
ECL^{i}PS^e—A Platform for Constraint Programming

Vol. 11 Iss. 2 – January 1997

The Year 2000 Problem
Working with Users to Generate Organisational Requirements:
The ORDIT Methodology
Network computing with remote Windows
Neural Networks
Short-term currency forecasting using neural networks
Helping Retailers Generate Customer Relationships
The Systems Engineering Excellence Model
Cochise: a World Wide Web interface to TPMS applications

Vol. 11 Iss. 1 – May 1996

The Internet and how it is used
An Architecture for a Business Data Warehouse
Virtual Reality as an Aid to Data Visualization
Re-engineering the Hardware of CAFS
An Innovative Solution for the Interconnection of Future Component Packaging
Development of Practical Verification Tools
Coupling ORACLE with ECLiPSe
Integrating the Object Database System ODB-II with Object Request Brokers
SAMSON and the Management of SESAME

Vol. 10 Iss. 2 – November 1995

The Architecture of the ICL GOLDRUSH MegaSERVER
The Hardware Architecture of the ICL GOLDRUSH MegaSERVER
CAL in Higher Education – Potential and Pitfalls
The UK Technology Foresight Programme
Making the Internet Safe for Business
Developing Financial Services Kiosks
High Availability Manager
The Virgin Global Challenger
Design of the Format for EDI Messages Using Object–Oriented Techniques
New Aspects of Research on Displays

Vol. 10 Iss. 1 – May 1995

Object databases and their role in multimedia information systems
The ICL Multimedia Desktop Programme
Multimedia Information used in Learning Organisations
The Software Paradigm
Single Sign–on Systems
Why is it difficult producing safety–critical software?
Experiences using the Ingres Search Accelerator for a Large Property Management Database System
RAID
Improving Configuration Management for Complex Open Systems

Vol. 9 Iss. 2 – November 1994

Establishing Co-operation in Federated Systems
An ANSA Analysis of Open Dependable Distributed Computing
An Open Architecture for Real–Time Processing
Updating the Secure Office System
POSIX Security Framework
SQL Gateways for Client-Server Systems
Asynchronous transfer mode – ATM
The ICL search accelerator™, SCAFS™: functionality and benefits
Open Teleservice – A Framework for Service in the 90s
LEO, A personal memoire

Vol. 9 Iss. 1 – May 1994

Client-server architecture
How ICL Corporate Systems support Client-server: an Architectural Overview
Exploiting Client-server Computing to meet the needs of Retail Banking Organisations
A practical example of Client-server Integration
From a Frog to a Handsome Prince: Enhancing existing character based mainframe applications
Legacy systems in client-server networks: A gateway employing scripted terminal emulation
The Management of Client-server Systems
Dialogue Manager: Integrating disparate services in client–server environments
Distributed Printing in a Heterogeneous World
Systems Management: an example of a successful Client–server Architecture
PARIS – ICL's Problem & Resolution Information System

Vol. 8 Iss. 4 – November 1993

Toward the 4th Generation Office: A Study in Office Systems Evolution
IPCS – Integrated Product Configuring Service
CGS – The ICL Configurer Graphics Service

Location Transparency in Heterogeneous Networks
Future Office Interconnection Architectures for LAN and Wide Area Access
Parallel Lisp and the Text Translation System METAL on the European Declarative System
Detecting Latent Sector Faults in SCSI Disks

Vol. 8 Iss. 3 – May 1993

An Introduction to OPEN*framework*
The Evolution of the OPEN*framework* Systems Architecture
Creating Potential for Change
OPEN*framework* in Action at DEVETIR
Strategic Information Systems planning: A Process to Integrate IT and Business Systems
Describing Systems in the OPEN*framework* Integration Knowledge Base
Multimedia and Standards for Open Information
VME–X: Making VME Open
A New Approach to Cryptographic Facility Design
CHISLE: An Engineer's Tool for Hardware System Design
Distributed Detection of Deadlock

Vol. 8 Iss. 2 – November 1992

Open Networks – The Key to Global Success
Infrastructure of Corporate Networks in the Nineties
Broadband Networking
FDDI – The High Speed Network of the Nineties
The Evolution of Wireless Networks
Communications Technology for the Retail Environment
RIBA – A Support Environment for Distributed Processing
Information Technology: Support for Law Enforcement Investigations and Intelligence
Standard for Keyboard Layouts – The Origins and Scope of ISO/TEC 9995
ESS – A Solid State Disc System for ICL System for ICL Series 39 Mainframes

Vol. 8 Iss. 1 – May 1992

Defining CASE Requirements
ICL's ICASE Products
The Engineering Database
CASE Data Integration: The Emerging International Standards
Building Maintainable Knowledge Based Systems
The Architecture of an Open Dictionary
The Use of a Persistent Language in the Implementation of a Process Support System
ALF: A Third Generation Environment for Systems Engineering
MASP/DL: The ALF Language for Process Modelling
The ALF User Interface Management System
A New Notation for Dataflow Specifications

Vol. 7 Iss. 4 – November 1991

Systems Management: A Challenge for the Nineties – Why now?
The Evolution within ICL of an Architecture for Systems Management
Manageability of a Distributed System
Distribution Management – ICL's Open Approach
Experience of Managing Data Flows in Distributed Computing in Retail Businesses
Generation of Configurations – a Collaborative Venture
Operations Management
OSMC: The Operations Control Manager
The Network Management Domain
An Overview of the Raleigh Object–Oriented Database System
Making a Secure Office System
Architectures of Knowledge Base Machines
The Origins of PERICLES – A common on–line Interface

Vol. 7 Iss. 3 – May 1991

Introduction to the technical characteristics of ISDN
ISDN in France: Numéris and its market
The Telecoms Scene in Spain
Future Applications of ISDN to Information Technology

A Geographical Information System for Managing the Assets of a Water Company
Using Constraint Logic Programming Techniques in Container Port Planning
Locator – An Application of Knowledge Engineering to ICL's Customer Service
Designing the HCI for a Graphical Knowledge Tree Editor: A Case Study in User-Centred Design
X/OPEN – From Strength to Strength
Architectures of Database Machines
Computer Simulation for the Efficient Development of Silicon Technologies
The use of Ward and Mellor Structured Methodology for the Design of a Complex Real Time Syste

Vol. 7 Iss. 2 – November 1990

The SX Node Architecture
SX Design Process
Physical Design Concepts of the SX Mainframe
The Development of Marketing to Design: The Incorporation of Human Factors into Specification and Design
Advances in the Processing and Management of Multimedia Information
An Overview of Multiworks
RICHE–Rèseau d'Information et de Communication Hospitalier Européen (Healthcare Information and Communication Network for Europe)
E.S.F – A European Programme for Evolutionary Introduction of Software Factories
A Spreadsheet with Visible Logic
Intelligent Help – The Results of the EUROHELP Project
How to use Colour in Displays – Coding, Cognition and Comprehension
Eye Movements for A Bidirectional Human Interface
Government IT Infrastructure for the Nineties (GIN): An Introduction to the Programme

Vol. 7 Iss. 1 – May 1990

Architecture of the DRS6000 (UNICORN) Hardware
DRS6000 (UNICORN) software: an overview
Electromechanical Design of DRS6000 (UNICORN)
The User–System Interface – a challenge for application users and application developers?
The emergence of the separable user interface
SMIS – A Knowledge–Based Interface to Marketing Data
A Conversational Interface to a Constraint–Satisfaction System
SODA: The ICL interface for ODA document access
Human – Human co–operation and the design of co–operative mechanisms
Regulatory Requirements for Security – User Access Control
Standards for secure interfaces to distributed applications
How to Use Colour in Displays – 1. Physiology Physics & Perception

Vol. 6 Iss. 4 – November 1989

Time to Market in new product development
Time to Market in manufacturing
The VME High Security Option
Security aspects of the fundamental association model
An introduction to public key systems and digital signatures
Security classes and access rights in a distributed system
Building a marketeer's workbench: an expert system applied to the marketing planning process
The Knowledge Crunching Machine at ECRC: a joint R&D project of a high speed Prolog system
Aspects of protection on the Flagship machine: binding, context and environment
ICL Company Research and Development Part 3: The New Range and other developments

Vol. 6 Iss. 3 – May 1989

Tools, Methods and Theories: a personal view of progress towards Systems Engineering
Systems Integration
An architectural framework for systems
Twenty Years with Support Environments
An Introduction to the IPSE 2.5 Project
The case for CASE
The UK Inland Revenue operational systems
La solution ICL chez Carrefour a Orleans
A Formally–Specified In–Store System for the Retail Sector towards a Geographic Information System
Ingres Physical Design Adviser: a prototype system for advising on the physical design of an Ingres relational database
KANT – a Knowledge Analysis Tool

Pure Logic Language
The 'Design to Product' Alvey Demonstrator

Vol. 6 Iss. 2 – November 1988

Flexible Manufacturing at ICL's Ashton plant
Knowledge based systems in computer based manufacturing
Open systems architecture for CIM
MAES – An expert system applied to the planning of material supply in computer manufacturing
JIT and IT
Computer Aided Process Planning (CAPP): Experience at Dowty Fuel Systems
Use of integrated electronic mail within databases to control processes
Value engineering – a tool for product cost reduction
ASP: Artwork specifications in Prolog
Elastomer technology for probing high-density printed circuit boards
The effects of back-driving surface mounted digital integrated circuits
Reliability of surface-mounted component soldered joints produced by vapour phase, infrared soldering techniques
Materials evaluation
On the human side of technology

Vol. 6 Iss. 1 – May 1988

ICL Series 39 support process
The ICL systems support centre organisation
ICL Services Product Centre
Knowledge engineering as an aid to the system service desks
Logic analysers for system problem solving
Repair – past and future
OSI migration
A Network to Support Application Software Development
Universal Communications Cabling: A Building Utility
Collecting and generalising knowledge descriptions from task analysis data
The architecture of an automated Quality Management System
ICL Company Research and Development Part 2: Mergers and Mainframes, 1959–1968

Vol. 5 Iss. 4 – November 1987

Open Distributed Processing
The Advanced Network Systems Architecture project
Community management for the ICL networked production line
The X/OPEN Group and the Common Applications Environment
Security in distributed information systems: needs, problems and solutions
Cryptographic file storage
Standards and office information
Introducing ODA
The Technical and Office Protocols – TOP
X400 – international information distribution
A general purpose natural language interface: design and application as a database front-end
DAP–Ada: Ada facilities for SIMD architectures
Quick language implementation

Vol. 5 Iss. 3 – May 1987

What is Fifth Generation? – the scope of the ICL programme
The Alvey DHSS Large Demonstrator Project
PARAMEDICL: a computer–aided medical diagnosis system for parallel architectures
S39XC – a configurer for Series 39 mainframe systems
The application of knowledge–based systems to computer capacity management
On knowledge bases at ECRC
Logic languages and relational databases: the design and implementation of Educe
The semantic aspects of MMI
Language overview
PISA – a Persistent Information Space Architecture
Software development using functional programming languages
Dactl: a computational model and compiler target language based on graph reduction
Designing system software for parallel declarative systems
Flagship computational models and machine architecture

Flagship hardware and implementation
GRIP: a parallel graph–reduction machine

Vol. 5 Iss. 2 – November 1986

The Management into the 1990s Research Programme
Managing strategic ideas: the role of the computer
A study of interactive computing at top management levels
A management support environment
Managing change and gaining corporate commitment
An approach to information technology planning
Preparing and organising for IPSE
Global Language for Distributed Data Integration
The design of distributed secure logical machines
Mathematical logic in the large practical world
The ICL DRS300 management graphics system
Performance of OSLAN local area network
Experience with programming parallel signal–processing algorithms in Fortran 8X

Vol. 5 Iss. 1 – May 1986

ICL company research and development, 1904–1959
Innovation in computational architecture and design
REMIT: a natural language paraphraser for relational query expressions
Natural language database enquiry
The *me too* method of software design
Formal specification – a simple example
The effects of inspections on software quality and productivity
Recent developments in image data compression for digital facsimile
Message structure as a determinant of message processing system structure

Vol. 4 Iss. 4 – November 1985

History of the ICL content-addressable file store, (CAFS)
History of the CAFS relational software
The CAFS system today and tomorrow
Development of the CAFS-ISP controller product for Series 29 and 39 systems
CAFS–ISP: issues for the applications designer
Using secondary indexes for large CAFS databases
Creating an end-user CAFS service
Textmaster – a document retrieval system using CAFS–ISP
CAFS and text: the view from academia
Secrets of the sky: the IRAS data at Queen Mary College
CAFS file-correlation unit

Vol. 4 Iss. 3 – May 1985

Overview of the ICL Series 39 Level 30 system
VME nodal architecture: a model for the realisation of a distributed system concept
Processing node of the ICL Series 39 Level 30 system
Input/output controller and local area networks of the ICL Series 39 Level 30 system
The store of the ICL Series 39 Level 30 system
The high–speed peripheral controller for the Series 39 system
Development of 8000–gate CMOS gate arrays for the ICL Level 30 system
Development route for the C8K 8000–gate CMOS array
Design automation tools used in the development of the ICL Series 39 Level 30 system
Design and manufacture of the cabinet for the ICL Series 39 Level 30 system
Manufacturing the level 30 system I Mercury: an advanced production line
Manufacturing the Level 30 system II Merlin: an advanced printed circuit board manufacturing system
Manufacturing the Level 30 system III The test system

Vol. 4 Iss. 2 – November 1984

Modelling a multi–processor designed for telecommunication systems control
Tracking of LSI chips and printed circuit boards using the ICL Distributed Array Processor
Sorting on DAP
User functions for the generation and distribution of encipherment keys

Analysis of software failure data(1): adaptation of the Littlewood stochastic reliability growth model for coarse data
Towards a formal specification of the ICL Data Dictionary

Vol. 4 Iss. 1 – May 1984

The ICL University Research Council
The Atlas 10 computer
Towards better specifications
Solution of the global element equations on the ICL DAP
Quality model of system design and integration
Software cost models
Program history records: a system of software data collection and analysis

Vol. 3 Iss. 4 – November 1983

Expert system in heavy industry: an application of ICLX in a British Steel Corporation works
Dragon: the development of an expert sizing system
The logic language PROLOG–M in database technology and intelligent knowledge–based systems
QPROC: a natural language database enquiry system implemented in PROLOG
Modelling software support

Vol. 3 Iss. 3 – May 1983

IPA networking architecture
IPA data interchange and networking facilities
The IPA telecommunications function
IPA community management
MACROLAN: a high–performance network
Specification in CSP language of the ECMA–72 Class 4 transport protocol
Evolution of switched telecommunication networks
DAP in action

Vol. 3 Iss. 2 – November 1982

The advance of Information Technology
Computing for the needs of development in the smallholder sector
The PERQ workstation and the distributed computing environment
Some techniques for handling encipherment keys
The use of COBOL for scientific data processing
Recognition of hand–written characters using the DAP
Hardware design faults: a classification and some measurements

Vol. 3 Iss. 1 – May 1982

Software of the ICL System 25
Security in a large general–purpose operating system: ICL's approach in VME/2900
Systems evolution dynamics of VME/B
Software aspects of the Exeter Community Health Services Computer Project
Associative data management system
Evaluating manufacturing testing strategies

Vol. 2 Iss. 4 – November 1981

Architecture of the ICL System 25
Designing for the X25 telecommunications standard
Viewdata and the ICL Bulletin System
Development philosophy and fundamental processing concepts of the ICL Rapid Application Development System RADS
A moving–mesh plasma equilibrium problem on the ICL Distributed Array Processor

Vol. 2 Iss. 3 – May 1981

A dynamic database for econometric modelling
Personnel on CAFS: a case study
Giving the computer a voice
Data integrity and the implications for back–up
Applications of the ICL Distributed Array Processor to econometric computations
A high–level logic design system
Measures of programming complexity

Vol. 2 Iss. 2 – November 1980

The ICL Information Processing Architecture, IPA
VME/B: a model for the realisation of a total system concept
Birds, Bs and CRTs
Solution of elliptic partial differential equations on the ICL Distributed Array Processor
Data routing and transpositions in processor arrays
A Bayesian approach to test modelling

Vol. 2 Iss. 1 – May 1980

Security and privacy of data held in computers
CADES – software engineering in practice
ME29 Initial Program Load: an exercise in defensive programming
Project Little – an experimental ultra–reliable system
Flow of instructions through a pipelined processor
Towards an 'expert' diagnostic system
Using Open System Interconnection standards

Vol. 1 Iss. 3 – November 1979

Meteosat 1: Europe's first meteorological satellite
An analysis of checkpointing
Statistical and related systems
Structured programming techniques in interrupt–driven routines
The content addressable file store – CAFS
Computing in the humanities
The data dictionary system in analysis and design

Vol. 1 Iss. 2 – May 1979

Computers in support of agriculture in developing countries
Software and algorithms for the Distributed Array Processor
Hardware monitoring on the 2900 range
Network models of system performance
Advanced technology in printing: the laser printer
The new frontier: three essays on job control

Vol. 1 Iss. 1 – November 1978

The origins of the 2900 series
Sizing computer systems and workloads
Wind of Change
Standards for open–network operation
Distributed computing in business data processing
A general model for integrity control

To order back issues contact

Chrissie Wilson

Group Technical Directorate
ICL, Cavendish Road, Stevenage, Hertfordshire, SG1 2DY
Telephone +44 (0)1438 786321
Email: Chrissie.Wilson@icl.com

or

The Editor, V.A.J. Maller
Email: V.A.J.Maller@lboro.ac.uk
or: Victor.Maller@icl.com

Abstracts back to Volume 1 and full papers back to Volume 9, Issue 1 are available at http://www.icl.com/news/sjournal

ICL Systems Journal

Guidance for Authors

Content

The ICL Systems Journal has an international circulation. It publishes papers of a high standard that are related to ICL's business and is aimed at the general technical community and in particular at ICL's users, customers and staff. The Journal is intended for readers who have an interest in computing and its applications in general but who may not be informed on the topic covered by a particular paper. To be acceptable, papers on more specialised aspects of design or application must include some suitable introductory material or reference.

The Journal will not usually reprint papers already published but this does not necessarily exclude papers presented at conferences. It is not necessary for the material to be entirely new or original. Papers will not reveal material relating to unannounced products of any of the ICL Group of companies.

Letters to the Editor and book reviews may also be published.

Authors

Within the framework defined in paragraph 1, the Editor will be happy to consider a paper by any author or group of authors, whether or not employed by a company in the ICL Group. All papers will be judged on their merit, irrespective of origin.

Length

There is no fixed upper or lower limit, but a useful working range is 4,000-8,000 words; it may be difficult to accommodate a long paper in a particular issue. Authors should always keep brevity in mind but should not sacrifice necessary fullness of explanation.

Abstract

All papers should have an Abstract of approximately 200 words, suitable for the various abstracting journals to use without alteration.

Presentation

Printed (typed) copy

A typed copy of the manuscript, single sided on A4 paper with the pages numbered in sequence, should be sent to the Editor. Particular care should be taken to ensure that mathematical symbols and expressions, and any special characters such as Greek letters, are clear. Any detailed mathematical treatment should be put in an Appendix so that only essential results need be referred to in the text.

Electronic version

Authors are encouraged to submit either a magnetic disk version of their manuscript or a compressed e-mail attached file or both. The format of the file should conform to the standards of any of the widely used word processing packages or be a simple text file.

Diagrams

Line diagrams will usually be redrawn and professionally lettered for publication, so it is essential that the originals are clear. Axes of graphs should be labelled with the relevant variables and, where this is desirable, marked off with their values. All diagrams should be numbered for reference in the text and the text marked with the reference and an appropriate caption to show where each should be placed. Authors should check that all diagrams are actually referred to in the text and that copies of all diagrams referred to are supplied. Authors wishing to submit drawings in electronic form should ensure that they are separated from the main text and, ideally, be in the form of EPS files. If an author wishes to use colour, then it is very helpful that a professional drawing package be used, such as Adobe Illustrator. PowerPoint diagrams, however, may be used to indicate the author's requirements.

Photographs

Authors who wish to include photographs in their papers should provide good quality slides or prints, not digital images unless they have been taken with top quality professional equipment. High resolution scanned images are also acceptable in tiff or jpeg format.

Tables

As with diagrams, these should all have captions and reference numbers. If they are to be provided in electronic form, then either a standard spreadsheet (Excel) should be used or the data supplied as a file of comma/tab separated variables. A printed version should also be supplied, showing all row and column headings, as well as the relevant units for all the quantities tabulated.

References

Authors are asked to use the Author/Date system, in which the author(s) and the date of the publication are given in the text, and all the references are listed in alphabetical order of author at the end; e.g. in the text: "...further details are given in [Henderson, 1986]" with the corresponding entry in the reference list:

> HENDERSON, P., "Functional Programming Formal Specification and Rapid Prototyping," IEEE Trans. on Software Engineering SE 12, 2, 241-250, 1986.

Where there are more than two authors it is usual to give the text reference as "[X et al ...]".

Authors should check that all text references are listed; references to works not quoted in the text should be listed under a heading such as Bibliography or Further reading.

Style

The Editor aims to maintain a consistent style in terms of structure, grammar and spelling. The Oxford English Dictionary and its companion works form the basic reference and the Editor will be pleased to advise authors upon request.

Referees

The Editor may refer papers to independent referees for comment. If the referee recommends revisions to the draft, the author will be asked to make those revisions. Referees are anonymous. Minor editorial corrections, to conform to the Journal's general style for spelling, punctuation or notation, will be made by the Editor.

Proofs, Offprints

Proofs are sent to authors for correction before publication, usually in electronic form, either as PDF or as output files from the production system used for the Journal: PageMaker, InDesign, Illustrator and Photoshop.

Copyright

ICL Systems Journal Spring 2000

ICL Group Technical Directorate
Cavendish Road
Stevenage
Hertfordshire
SG1 2DY