

# ***The Power of Content***

*Achieving Competitive Advantage with  
Enterprise Content Management and TrueXML™*

## AN INTERLEAF WHITE PAPER

March 1999  
Barry Briggs

**Interleaf**

62 Fourth Avenue  
Waltham, MA 02154

781-768-1578 ☎  
781-290-4955 📠

[www.interleaf.com](http://www.interleaf.com)

## **Communicating at Light Speed**

Companies run on documents. Documents communicate product details, describe new innovations, carry bids and quotes, confirm terms, and handle payment. Documents teach customers how to use products, documents provide informative analysis, they seal agreements between companies, and between companies and customers. Documents are the foundation of business.

In the last generation computing has utterly and irrevocably changed the nature of documents. Only a few short decades ago a prospect desiring a product brochure would write, or perhaps telephone the vendor. A customer representative would then send a clerk to find the appropriate document lying, perhaps, on a shelf in a back room with all the other brochures, manuals, price lists, and warranties. Then the clerk would pack the brochure into an envelope, affix a stamp, and, hopefully within a week or two, the customer would have the brochure.

All that takes less than a few seconds today. Go to the company's home page, click a few times, and within moments the brochure is rendered before you on your computer screen.

Not since Gutenberg has the very nature of writing and communications changed so dramatically. Thanks to the near-meteoric rise of global computer networking, companies can reach suppliers and customers around the world, twenty-four hours a day, in the blink of an eye. The opportunities in this new, but demanding, business environment are exciting and limitless, as the reach of the Internet extends to very nearly every person on the planet.

It has only been a few short years since the Internet has revolutionized business. Yet in that time customers have already become accustomed to and expect current, up-to-date information to be available on companies' Web sites. Such expectations place extraordinary demands on content suppliers who must assure the currency, accuracy and thoroughness of their information.

Indeed, the Web is only one, if the most visible, manifestation of this new phenomenon which we call "just-in-time information." Everywhere product cycles are accelerating; what used to be require months of development now only takes days, and sometimes only hours. Even something as minutely complex as a custom computer chip can now be designed in a week, and be in production shortly thereafter; yet it is useless if its specifications cannot be simultaneously distributed to potential customers.

Moreover, customers are rapidly becoming accustomed to information designed specifically for their particular needs. Information systems must be able to produce customized data in real time for each of potentially millions of clients and partners. And, as electronic commerce networks grow more interdependent, the demand customized information transfer between corporate computing networks – e.g., parts catalogs – will continue to grow.

All of these demands will pose daunting challenges to IT professionals who must respond effectively to accelerating requirements for instant information production. In the remainder of this white paper we will describe Interleaf's solution to this problem, which we call *enterprise content management*.

## The Solution: Enterprise Content Management

What is needed is an enterprise information infrastructure; we call this the *enterprise content management (ECM) backbone*. ECM is a technology which, by instantly responding to the needs of all of a company's information consumers, propels the enterprise into new dimensions of efficiency, productivity, and customer loyalty.

What is enterprise content management, and how does it work? Concisely, ECM consists of technologies aimed at dynamically creating and bundling together information "chunks" to form coherent packages, such as documents, manuals, publications, or Web pages. The key word in this definition is "dynamic" – for until a few years ago, a typical document or publication simply consisted of words and images on paper. It usually required weeks if not months to create, and needed the combined talents of writers, illustrators, layout designers, typesetters, and so on.

Such documents were hardly dynamic; once printed, they never changed.

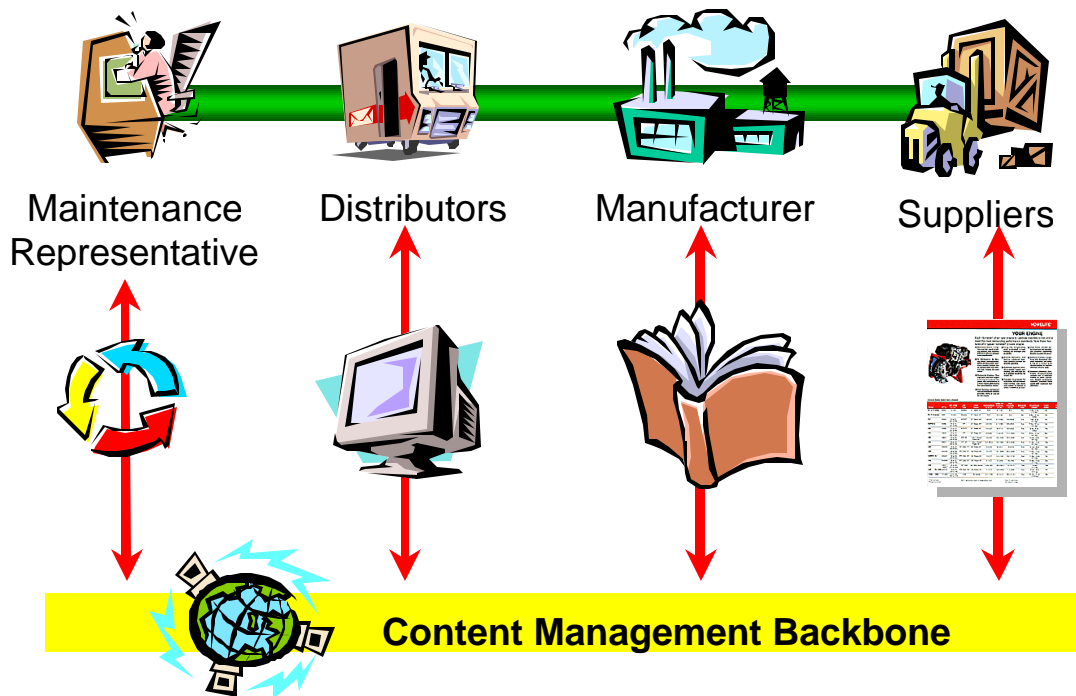


Figure 1. Content Management Infrastructure

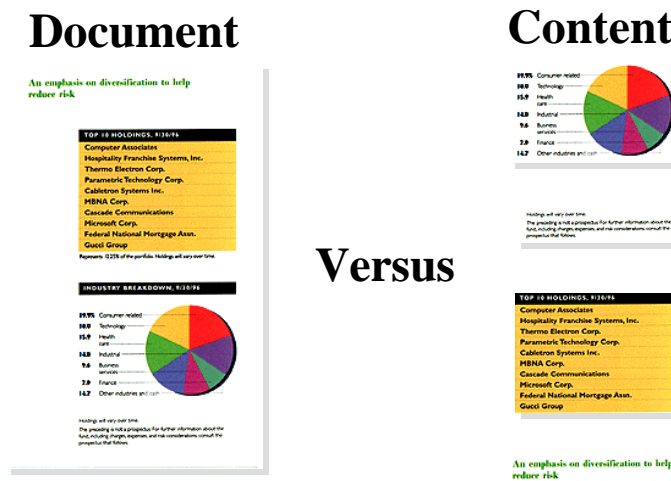
But as we have seen, today's requirements for information have made documents of this sort very nearly obsolete. Consider an electronic commerce environment in which goods are sold via a Web site. Descriptions of these goods are posted daily; in many cases the descriptions consist of some amount of shared information – mailing addresses, legal

disclaimers, contact information, discount rates – and some amount of variable information. Further, the information they present may be customized based upon previously known customer preferences.

For example, a service manual for a user’s computer might consist of information relating to the particular model of motherboard, the video card, the sound card, the monitor, and so on. What is particularly interesting is that the service manual as a whole might never really exist, but rather be dynamically assembled from records describing the user’s configuration.

This is the essence of a content management system: that information is treated as granular, manageable “chunks,” or *components*, which can be accessed, updated, and combined in real time. Thus, to build a document, the content management system might be queried to locate part information, the customer’s discount rate, the appropriate legal disclaimer, and so on; and then the document would be rendered to any number of output media.

Figure 2. Document Composed of Content Components



One key feature of ECM systems is that the content “chunks” as we’ve described them consist only of data – all presentation information is kept separate. This allows the *styling* to occur as the information is being delivered, and this in turn means that information managed by an ECM system can be easily repurposed to any number of targets. For example, since the information is managed, it can be easily and automatically reformatted for devices such as handheld computers and pagers.

Properly structured, information in an ECM system can represent a significant asset to the corporation. Because the data is kept in a semantically rich database, powerful search engines can rapidly scan through the data and return useful, relevant information, far more useful than typical present-day Web searches.

Moreover, again because of its structure, the ECM data can be culled by data mining and other business intelligence tools in order to extrapolate key trends and other hidden data – again providing distinct competitive advantage.

Search and data mining, of course, are two key foundations for the emerging science of knowledge management. KM seeks to structure and make explicit all the knowledge embedded in a corporation. A well-designed enterprise content management backbone supplying information in real time to a wide audience of consumers is indeed the basis for knowledge management.

## Requirements of an Enterprise Content Management System

The enterprise content management system can thus be seen to be the *foundation* of a company's information infrastructure. Indeed, the more complete its implementation, the more value it provides its users. Let us examine in particular the feature set an ECM should optimally provide:

- *End to end content control.* Ideally the ECM should be able to control and manage content from its creation, that is, its authoring, through the document assembly process and ultimately to publishing.
- *Management of document components.* As we have seen, modern documents are composed of components, which are assembled dynamically. Each of these components should be securely managed; that is, the ECM should require editors to "check out" components (much as they would a library book) in order to prevent concurrency issues. Moreover, the ECM should provide maintain multiple versions of a component so that prior versions can be accessed if desired.
- *Offline operation.* In today's mobile world knowledge workers are frequently disconnected from the corporate network. This limitation however should not impede their ability to work on ECM-managed documents or components.
- *Popular document creation tools.* The ECM should not require an entirely new investment in knowledge worker training; rather, it should support popular tools, such as Microsoft Word™ in addition to others.
- *Powerful, content-centric database.* Traditional table-oriented relational databases do not map well to content stores. Why? Because documents tend not to have the formal, strict structure required; mapping them to rows and columns can be cumbersome and slow. Rather, the store should easily accommodate the flexible, semi-structured nature of documents.
- *Dynamic assembly.* As we have seen, Just-in-Time documents are created in response to a consumer's query, which might be as simple as a Web request. This means that the document, which might never have existed before as a single, consolidated unit, might to satisfy this request be built for the first time for that customer.
- *Separate style from content.* Style (fonts, colors, margins, and so forth) should be kept separately from the actual content. Why? So that the content can be directly to as wide as possible number of targets.
- *Powerful publishing engine.* No content management system is complete without a means of publishing its documents. The publishing engine should permit documents of any size, ranging from a page or two to thousands of pages, to be rendered quickly

and efficiently. Moreover, the engine should be flexible enough to support a wide variety of formats, from the Web's Hypertext Markup Language (HTML) to PostScript, PCL, and many others for print. In each case the publishing engine should style the document appropriately for the output medium.

- *Open standards.* An ECM should support open, published standards. From a technical point of view open standards allow easy interoperability between machines and formats; from a business point of view open standards supply investment protection.

## Structuring and Formatting Content

In order to realize the potential afforded by an Enterprise Content Management system, content is semantically marked so that it can be searched upon and accessed quickly. For example, textual components describing manufacturing plants might use the word “Manufacturer” as a flag or index into the content database.

In fact, so-called “markup languages” have been in existence for some time. Until recently the most prevalent was SGML, the Standard Generalized Markup Language. SGML used textual “tags” to describe content. Thus a fragment of SGML providing information about a particular manufacturer might look like this:

```
<manufacturer >  
<name>Tyrell Corporation </name>  
<street-address>14 Main Street</street-address>  
<city-address>Anyplace, USA</city-address>  
</manufacturer >
```

*Figure 3. Adding Structure to Text*

The markers, or tags, in SGML are the data inside the angle brackets (< and >).

This seems very powerful; yet SGML usage, while successful in certain markets, failed to become widespread because it, and the styling language associated with it, were extremely complex and difficult to implement. Yet SGML directly led to one of the most important developments in the history of computing.

Of course, experienced Internet users will recognize the syntax of the SGML example above as very reminiscent of the language of the World Wide Web, Hypertext Markup Language, or HTML. In fact HTML, now enormously popular, is a much-simplified dialect of SGML.

The growth and penetration of HTML certainly are extraordinary. However, most experts now agree that HTML has serious deficiencies. Most important of these is that HTML blends style information with content; that is to say, information concerning the *presentation* of the data is contained along with the data itself.

Thus a (very simple) Web page for Tyrell Corporation might look like this:

```
<html>
<title>Tyrell Corporation</title>
<h1>Robotic Devices</h1>
<body><center>
14 Main Street
Anyplace, USA
</center></body>
</html>
```

Figure 4. Simple HTML Web Page

Note that in this short (but perfectly valid) HTML there are no *semantic* markers, that is to say, nothing indicating the *meaning* of the data. Rather, HTML describes how the text should be presented: as a heading, centered, as the title, and so on. Unlike SGML, HTML carries with it no knowledge of what the data is actually about. Consider then a database full of Web pages about robot manufacturers, patterned after the example above. It would be very nearly impossible to formulate a query of such data to “find all manufacturers in Anyplace, USA” – the HTML has no concept of a manufacturer.

It is nearly certain that HTML, given its pervasiveness, will remain for a long time. However, we must recognize that its value is primarily that of an output format. What is needed is an intermediate language that will allow us to describe and manipulate content.

## Enter XML

In 1996 the World Wide Web Consortium (W3C) and some 80 SGML experts joined forces to develop a permanent solution to the problems of HTML. After much design and debate, the eXtensible Markup Language, or XML, was standardized in February of 1998.

XML:

- Separates, as does SGML, content from style;
- Is, unlike SGML, easy to implement, making it more likely that low cost XML tools will be available;
- Has broad industry support from such industry leaders as Microsoft, IBM and Sun Microsystems.

Like SGML, XML provides meaning through use of tags. Hence, the SGML example we used above:

```
<manufacturer>
<name>Tyrell Corporation</name>
<street-address>14 Main Street</street-address>
<city-address>Anyplace, USA</city-address>
</manufacturer>
```

*Figure 5. XML Fragment*

is perfectly valid XML as well. Consider now how easy it is to envision a database query for all manufacturers in Anyplace.

As with SGML, one can create an application-specific dialect of XML – in concept not unlike a database schema. These dialects are defined by Document Type Definitions, or DTD's. For example, the World Wide Web Consortium has created DTD's for mathematical expressions; the resultant syntax is called MathML, and it is a dialect of XML.

More importantly, customers can build their own DTD's for specific applications. Thus an electronics company might create a DTD describing integrated circuits, or an online bookseller might have a DTD for its books.

In many cases, for business-to-business transactions, it will be necessary for companies to agree on the format of the data being exchanged. Until recently, EDI (Electronic Data Interchange), a language for invoices, billing, and so forth, was that language; today EDI is in large part being supplanted or augmented by XML, and this effort has wide industry support.

## Adding Style with XSL

Of course, in order to publish any XML document to a human-readable form, it must first be styled. To that end the W3C is presently working on the eXtensible Styling Language (XSL), a powerful mechanism which can transform XML documents into any number of formats, including HTML, PostScript, or even raw ASCII text. The standardization process for XSL is expected to be completed by the summer of 1999.

How does XSL work? Essentially it *transforms* XML according to its internal rules; different XSL files might transform the same XML file into (say) HTML, PostScript, or even raw ASCII format. In this way XSL enables the same *content* (XML) to be retargetted to any number of output media.

In the following example two fragments, one of XML and the other, XSL, demonstrate how the two are merged to create final output format. The details of how XSL works are beyond the scope of this paper; but observant readers can recognize how XSL matches styling to particular content tags.

```
<?xml version="1.0" standalone="yes"?>
<title>Robot and Android Vendors</title>
<manufacturer>
  <name>Tyrell Corporation</name>
  <rating>***</rating>
  <description>The most lifelike androids on the market come from
Tyrell Corporation. Extremely difficult to distinguish from naturally
generated organisms; however, a short lifespan
limits their utility to mining and other activities on other worlds .
</description>
  <photo>Tyrell.jpg</photo>
</manufacturer>
<manufacturer>
  <name>Asimov Inc</name>
  <rating>***</rating>
  <description>The inventor of the android concept brings artificial
robotic technology to new heights with its latest models including an
advanced detective model.
</description>
  <photo>asimov.jpg</photo>
</ manufacturer >
```

Figure 6. XML File for Robot Manufacturers

```

<xsl:template match="manufacturer">
  <fo:block font-family="Times New Roman"
    font-style="italic"
    font-size:12pt">
    <xsl:apply-templates/>
  </xsl:template>
<xsl:template match="name">
  <fo:block font-weight="bold">
    <apply-templates/>
  </xsl:template>
<xsl:template match="rating">
  <fo:block font-family="Times New Roman">
    <apply-templates/>
  </xsl:template>

```

Figure 7. XSL Fragment

## Robotic Manufacturers

### *Asimov, Inc.* --- \*\*\*\*

The inventor of the android concept brings artificial robotic technology to new heights with its latest models including an advanced detective model.



### *Tyrell Corporation* -- \*\*\*

The most lifelike androids on the market come from Tyrell Corporation. Extremely difficult to distinguish from naturally generated organisms; however, a short lifespan limits their utility to mining and other activities on other worlds.



### *Android International* --- \*\*\*

Super-expensive product with highly versatile and interchangeable limbs make this product a pleasure to review. Better lifespan than Tyrell products but lacking in emotional response.



Figure 8. Resultant Output (HTML as rendered in a browser)

Interestingly, XSL can actually filter and transform the XML data. Thus, another XSL style sheet applied to the *same* XML data might yield the following output:

<b>Robot Manufacturers</b>	
<b>Name</b>	<b>Rating</b>
Asimov	****
Tyrell Corporation	***
Android International	***
Bots Inc	***
Homunculus Inc.	**
Advanced Humanoids	**

*Figure 9. Alternate Rendering of XML*

Thus XSL can be used not only to format output for different devices and media, but can actually control what is included in the output. We at Interleaf believe XSL delivers tremendous value to our customers, as it is an extraordinarily powerful mechanism for operating on data, and, as we shall shortly present, we have made a significant investment in XSL.

## End to End Content Management: Interleaf's BladeRunner

Interleaf's content management product, code-named BladeRunner, is the most comprehensive and full-featured ECM system available on the market today.

Let us examine each of BladeRunner's features by following a document's life cycle as it is created, managed, and published. The following diagram depicts this cycle, and is described in the following sections.

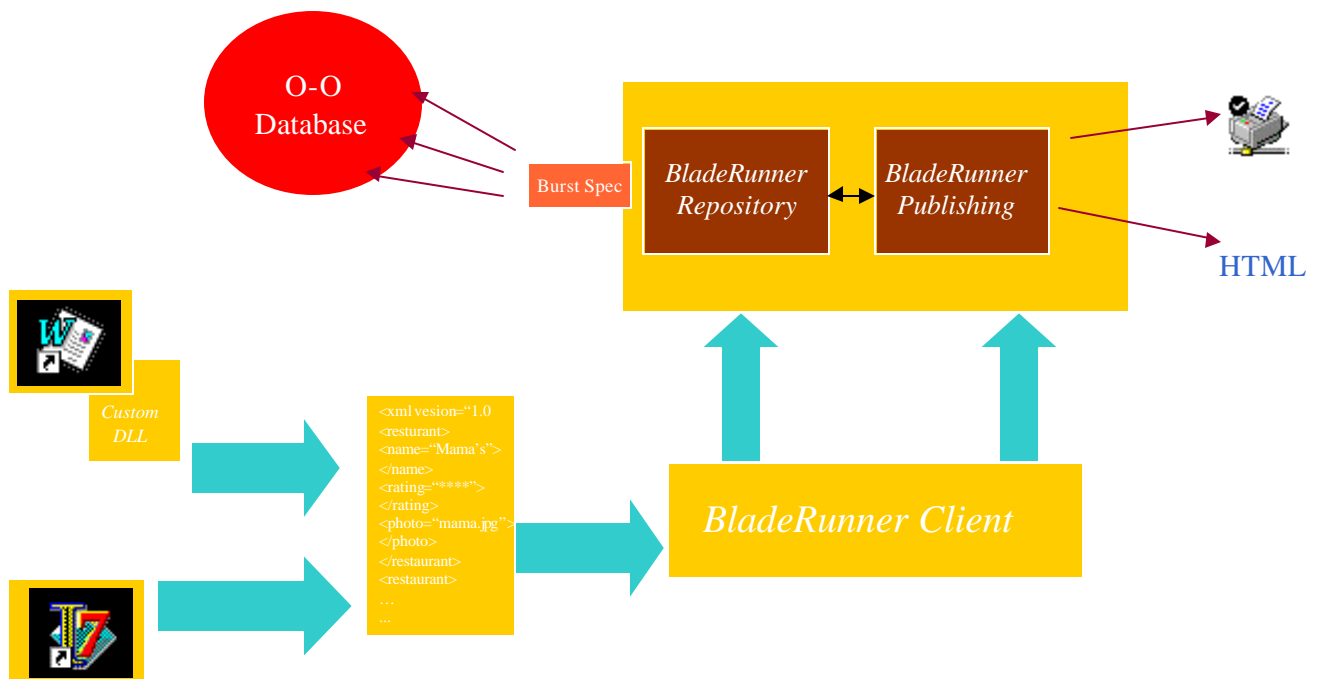


Figure 10. Document Flow through BladeRunner Content Management System

### Content Creation

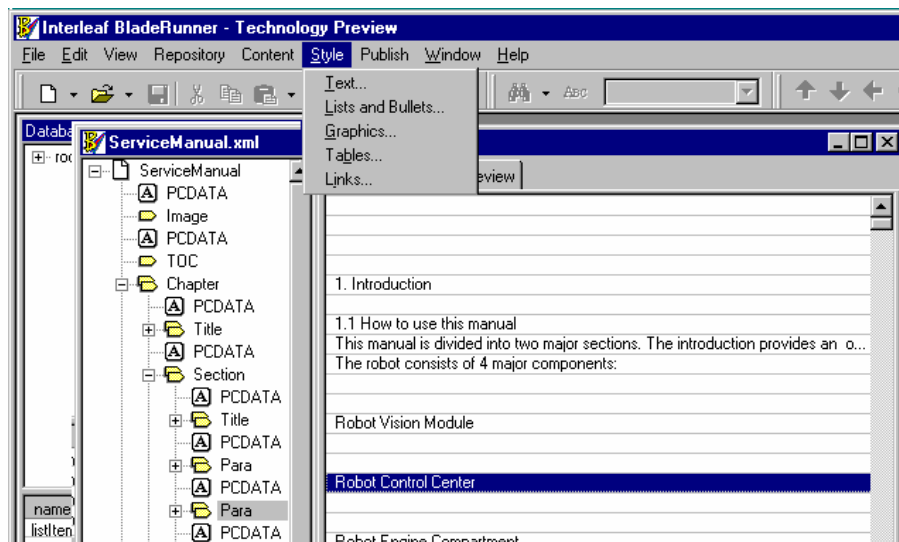
Authors, using a wide variety of tools, create structured content. One of the most popular content-creation tools is Microsoft's best-selling word processor, Word. Interleaf has built a Word addon that enables it to read and write XML. Because our implementation is designed to generate absolutely valid XML, we call it TrueXML.™

Any number of other tools can also be used. For example, Interleaf's high end complex SGML publishing product, Interleaf-7, is being enhanced to read and write XML for use by the BladeRunner ECM. Many other third parties are producing XML tools as well.

## *BladeRunner Composer*

The BladeRunner Composer is the vehicle through which XML and its reusable units are edited and searched, and output presentation and style can be defined. To navigate through documents and their constituent parts, the Composer begins with a *repository view*, a tree-structured depiction of data on the server. Users will find the Explorer-like interface familiar and easy to use.

The Composer allows users to check out XML and to perform operations on it. Users can define subsets of XML called *manageable units* that represent meaningful pieces of information. For instance, a six-step process to repair an assembly would be identified as a single reusable unit, as each of the six steps is required to complete the repair. Similarly, an address – consisting of a number, a street, a city, and a state – might better be handled as a single entity rather than as multiple, smaller units of information. These units are entirely under the control of the user.



*Figure 11. BladeRunner Composer*

If the user needs to change the ordering of elements within the document, the Composer's convenient drag-and-drop feature allows quick and easy changes.

The Composer can also be used to preview the document layout. Preview allows a user to see how a document will be rendered in any of the BladeRunner supported formats.

Within the Composer the Style Editor serves as the gateway to the publishing process through the creation and application of XSL style sheets. XSL is applied to the valid XML document that was created with the Composer to produce a rendering for the

appropriate publishing medium. The Style Editor uses an attribute list system to guide the creation of the style sheet. The current state of the document can be reviewed at any point using the preview feature.

Together, the Composer and Style Editor provide tools for the flexible and easy reuse of content. Rich, complex relationships between content are maintained through a sophisticated link management system while content layout and presentation is executed through a simple drag-and-drop interface.

### *Repository*

In order to store, maintain and manage content, Interleaf has created a content repository using an object-oriented model database. The implementation of the repository is both innovative and efficient, and features the ability to “burst” an XML document into its constituent parts.

A document as we have seen is composed of components. The repository understands the componentization, and can decompose and recompose documents in real time. Controlling the bursting process is a schema-like specification which describes to the system how documents of specific types should be constructed. For example, a description of real estate might be burst into description, address, and price components. This is entirely under the control of the document designer, and can be at virtually any level of granularity.

Once in the database, content is stored and managed with version control. Check-in and check-out algorithms eliminate concurrency issues. Users can search on text, content, properties, structure and meta-data (where used). The repository also manages links and references to different information units.

### *Publishing*

The publishing component takes the XSL and XML document and produces a rendering. With a batch publishing engine, Interleaf’s powerful assembly engine, and a dynamic publishing engine, Interleaf’s ECM system is capable of turning content into printed documents, CD, or Web content. Through the assembly engine, content may be bound in a persistent form to make documents or CDs, or it can be linked and bound just prior to delivery for HTML documents using Cascading Style Sheets. And any combination is possible: updates to a parts catalog might be posted daily to the Web and printed monthly. No matter what format is required, the reusable units are available to be reused as often and in as many different formats and combinations as necessary.

With this technology, customer-specific documents can be delivered real time via the Web or can be pressed on CD or printed. Each form can have its own presentation style, optimized for these selected media.

## **Building on a Content Management Infrastructure**

Interleaf's BladeRunner is far more than a product: it is a platform upon which we intend to layer many new and exciting technologies, giving customers yet more value as they enter the world of eBusiness. Among these of course are the many important standards now under discussion at the World Wide Web Consortium (W3C), including XML Namespaces, XML Schemas, XML Query Language (XQL), and XML Fragment Interchange, to name but a few. Interleaf has representatives on most of the relevant committees.

Interleaf is also committed to its vision of dynamically assembled documents, which we call "virtual documents." We intend to make it easy and straightforward to build systems which not only build documents from XML components, but can also reach out to other data stores such as corporate relational databases (DBMS's), file systems, and, indeed, special purpose storage systems such as video databases to create these documents. Interleaf will provide customers with the tools they need, including expert consulting, to build these robust systems.

Further, as search and data mining technologies continue to grow in sophistication, Interleaf will build these advanced capabilities into BladeRunner, again adding value to for its customers. Moreover, Interleaf will shortly ship a developer's Software Development Kit for BladeRunner allowing knowledgeable users to add extensions to both the client and server components of the product.

## Conclusions

The demands for just-in-time information will continue to grow at an accelerated pace through the next decade. Companies embracing technologies to enable production of such information will find themselves possessing competitive advantage in the marketplace.

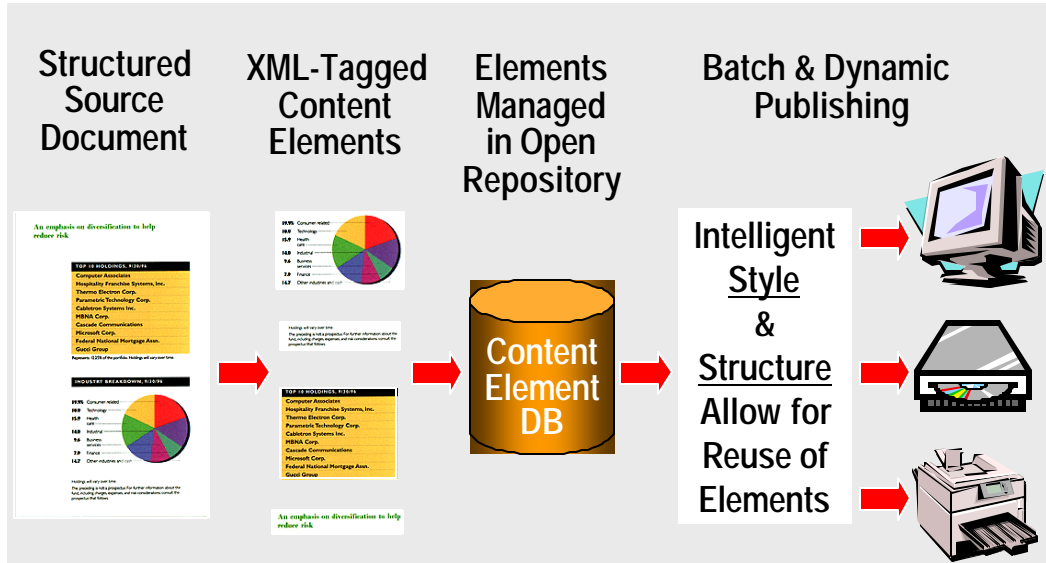


Figure 12. Power of Content Management

We at Interleaf believe that the BladeRunner product represents the most advanced implementation of an enterprise content management system. Its thoughtful and innovative use of XML gives it power and flexibility while not sacrificing performance.

BladeRunner leverages the tools most familiar to content creators, most notably Microsoft Word. This feature permits an easy and painless migration from legacy file based management or document management systems to content management. Further, by its industry-leading implementation of the eXtensible Styling Language (XSL) BladeRunner at a stroke makes content more powerful by enabling its publication to a variety of a target formats.

BladeRunner leverages its long history of complex publishing in its high performance server-based publishing engine, supporting everything from HTML to PostScript and many others.

Finally, alone in the market, BladeRunner provides an *end to end* solution for content management, from creation through publishing. We at Interleaf believe it is the platform for your success.