newspaper techniques

Ifra News. Faster, better and cheaper – that was the common theme during the recent Ifra Italia Conference.

Marketing. The British daily The Guardian and Xerox join forces to embark on a pilot scheme for print-on-demand.

Newsroom. Ifra's Kerry Northrup sits in on an internal course about News Disciplines at Tribune Company.

Prepress & Integration. Colour Management Live offers a new twist to Ifra symposia, and reaches practical solutions.

Electronic media. A new study by the Poynter Institute in America reveals that readers on the Internet

In the July/August 2000 edition of "newspaper techniques", the cover story was dedicated to "Building a powerful publishing platform." One of the articles published was entitled "Components, CORBA, EAI – keys to a complete publishing system?" The article was an edited version of a comprehensive report prepared by: Glen B. Alleman, Niwot Ridge Consulting (www.niwotridge.com) an associate of Nilan Sanders Associates (www.nilan-sanders.com) for System Integrators Inc. (www.sii.com). For more on the report, contact Alleman via e-mail: galleman@niwotridge.com.

Reprint from newspaper techniques, July/August 2000

Building a powerful publishing platform

Components, CORBA, EAI – keys to a complete publishing system?

Nowadays we are seeing a number of technologies emerging in the construction of software for the publishing industry. One technology that is gaining interest from the publishing world is the use of component technologies, with CORBA and EAI as the means of assembling these components into a complete system. This a summary of a presentation given at a recent IPTC meeting.

It's the value of information and technology assets that drives business decisions. How can the value of these assets be increased in the presence of changing standards, increasing complexity, increasing competition, and the Internet? One answer is the better use of commercial off-the-shelf components and reuse of these components once they have been integrated into a system. The Internet age has brought change to nearly every industry that uses software. In publishing the Internet means multiple media, new standards, and new architectures for the underlying systems.

The economics of the Internet Age mean an increasing returns business strategy, instability of markets, multiple outcomes, unpredictability, and the ability to lock a market. Adapting software to today's ever-shifting economy requires the same processes as adapting a business: flexibility and the ability to operate in the presence of chaos. This environment requires adaptability moreso than optimisation. Adaptability means agility, flexibility, incrementalism, rapid development, interfaces that are not formally defined, chaotic business and product environments, managing the state of the business rather than the flow of the business. Believers in component software say these systems provide the tools and architecture to meet these requirements, and that traditional software tools and architecture do not.

There are a lot of terms here, EAI, CORBA, components, so here is a brief explanation of how they each work together. We will go into greater detail later. A component-based system is composed of objects that interact through a standard interface protocol. The connections between the applications of this system are mediated by CORBA (Common Object Request Broker Architecture) in a platform- and programming language-independent manner. EAI (Enterprise Application Integration) is the overarching concept of deploying these components, where CORBA is the enabling technology, forming one of the foundations for the component technology used to implement EAI.

Why component-based systems?

Why are CORBA component-based systems different and better than traditional integrated systems? How can the user understand the differences and make an informed decision about the technology strategies needed to move into the next generation of publishing systems?
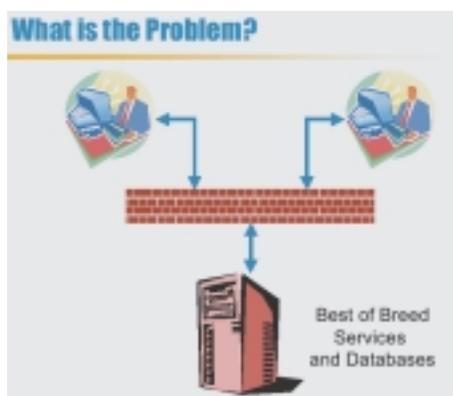
In the traditional approach to assembling a system from components, the individual API's (Application Programmable Interface), data formats, protocols, languages, operating systems, and application locales create many barriers to the "seamless" construction of the system. Once these connections have been made, any changes in any component in the system will have some impact on other components.

In the past, these brick walls have been overcome with brute force, either through the efforts of programming or the use of intermediate database or format translation software. This approach usually resulted in client-server architectures that were "fixed" in their construction, just like the brick wall. With the introduction of "middleware" interface, locality, programming languages and operating systems issues become the responsibility of CORBA. Using component technologies, the application domain and the services domain can be connected with very little impact on the other domains. This is done through CORBA's standard distributed object architectures.

Ins and outs of CORBA?

CORBA has several advantages, and some disadvantages, over the alternatives. It provides the services for fault tolerance, scaling, quality of service, replication, and its proponents say the technology enables:
> best of breed application components to interoperate;
> distributed heterogeneous systems integration to take place in a transparent manner;

Left: In the traditional approach to assembling a system from components, the connections between the various data formats, protocols, languages, operating systems, etc., created many barriers or "brick walls" to the seamless construction of the system. Right: By using CORBA, when changes are made to these different objects in the system, there is very little impact on each object.
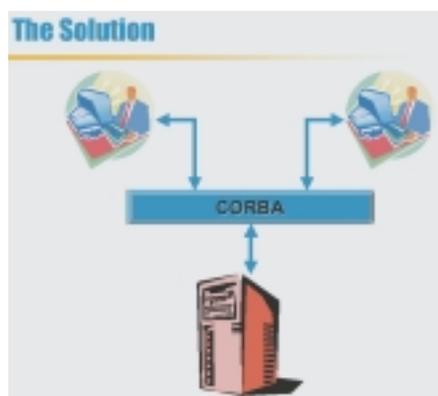
> legacy system integration strategies to be implemented in the same manner as new development;

> rapid development strategies using standard interfaces;

> Internet-based strategies using Java;

> and platform- and language-independent strategies.

These features come with a price, like anything worthwhile, but they are provided in a standard package. The price is the technical competence and skills needed to assemble the system into a product. In the current business environment these skills are in short supply.

How EAI applies?

In the publishing world the content of the publication is usually located in a variety of systems. Proponents of a component-based system say the days of a single monolithic system that does everything for everyone are gone. Today's systems are in fact systems of applications, each with a special capability. The users of this system want each of these specialised systems to work together seamlessly without requiring any changes to the underlying data or processes in each domain.

Like we said before, EAI is the overarching concept of deploying components. A more specific definition of EAI is "the integration of applications across the enterprise in a rational and centralised manner." The transparent interoperation of commercial off the shelf (COTS) products is what distinguishes an EAI solution from the traditional integrated solution. As we said, CORBA forms one of the foundations for the component technology used to implement EAI, but EAI results in a seamless integration of a variety of system applications to form a meta system. There are others of course, but in the publishing world EAI is an emerging concept. With the introduction of ERP systems, the connection between content creation and publishing and the business side of this process can be made.

Demands of the market should drive technology

Many of today's technology decisions are driven by the market, not the vendor. The Internet, CORBA, communications networks, programming languages, databases, system architectures, are all defined to the business needs of the user. Adapting to these standards requires an architectural vision, not just a product plan. This vision must include the evolving standards as well as the evolving component technologies provided by external suppliers. Building "purpose built" applications moves away from this vision and must be considered carefully.

We've heard it all before: Change is constant. Change is disruptive. Change is the engine that drives innovation, market share, product displacement, product obsolescence. In order to survive in the presence of change, the underlying system architecture must not only support change but also must thrive on change – that is, the cost of change is not related to the size or complexity of the application domain, but is a function of the change. This is a linear change cost model, rather than a non-linear (exponential or greater) models found in API-based implementations.

The traditional processes of analysis, design, development, and deployment have difficulty dealing with rapid change, adaptive markets, and changing market requirements. Component-based systems provide the means to develop products in the presence of these forces.

Out of the box

In order to design, develop and deploy component-based systems, new skills are needed. These include Object-Oriented analysis and design, formal system architecture, understandings of fault tolerance, scaling, and n-tier distributed processing. These skills are the foundation of the Internet. Searching for a product and vendor with these skills is a critical success factor in the acquisition of any modern system.

The Manufacturing Paradigm
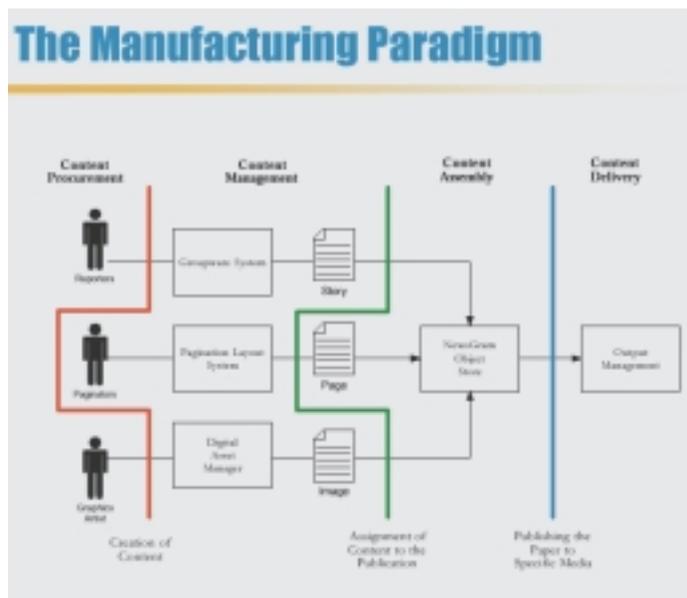
Here's a simple picture of the business processes involved in publishing a newspaper. The content creation and management within each domain can be provided by a best of breed application. The connectivity between the domains is provided through a layered set of objects which reside in a central clearing house. The state and status of the content in each domain is communicated to other domains using this meta object. The content remains in the application domain until it is assembled on the published page for output.

The alternatives to component-based systems can be found in the market place. They include traditional client-server based systems, best-of-breed applications connected to centralised databases, and tightly integrated applications using programmatic interfaces. Opponents say these fail to address the fundamental issues with modern software systems such as adaptive configurations, isolation of functions, layering of components, heterogeneous integration strategies, and standards for integration.

As usual, change along these lines in the publishing world is coming somewhat late compared to other businesses. This is an advantage that can be used to avoid many of the past difficulties.

CORBA factors

The following are some critical success factors to consider when thinking of CORBA:

1)	Adaptive platforms: Unix and NT dominate here, but there are multiple flavours of Unix as well as databases, and there is always the Mac.

2)	Agile application selection: the best of breed list changes with every generation of product. >

3) Scalable resources: scalable in terms of performance, capacity, users, locations.

4) Robust (fault tolerant) architectures: Tandem-like behaviours are the benchmark.

5) Natural transition to the Internet: CORBA and Java are "the natural path to the net," anything less will become an "Instant Legacy" system.

These critical success factors could be applied to any system integration architecture. So what makes CORBA unique here? Proponents say CORBA provides these in one package, external from the business applications.

It provides them through a set of standard interfaces, a standard processing model, and a standard set of platforms and programming languages.

Media companies using CORBA

A number of media companies have used CORBA in their various workflows. Here are a few.

CNN.com, a division of Cable News Network that is responsible for non-broadcast dissemination of news for CNN and provides content to more than 150 co-branded sites, realised several benefits through CORBA.

The company has been able to: build both the clients and servers simultaneously, while connecting them quickly and seamlessly, using IDL (Interface Definition Language); shorten its development cycle through Object Reuse and save time and money by minimising the debugging of integrated software.

deutsche presse agentur (dpa), the press agency based in Germany that provides multimedia information facilities for its correspondents and customers, created true multi-media operations with the help of CORBA and integrated new production systems with legacy libraries and workstations.

Britain's independent broadcasting company sky.com says its associated organisations "are building more powerful web-based solutions to deliver content. To build these applications, developers need to knit together their existing IT infrastructure with the latest web technologies."

Newspapers are sure to get in on the mix, but which path they choose is anyone's guess. <

Ifra, the world's leading association for newspaper and media technology