Network Publishing Paradigm:
A Web Authoring and Publishing Methodology for Internet Commerce

Benjamin Koo[ξ]
Mustafa Poonawala[φ]
Wei-Tek Tsai[φ]
Elim Qiu[ξ]
Editing and graphics by Jerry Schlei

[ξ]Integrity Solutions Inc.
Eagan, Minnesota.

[φ]University of Minnesota
Minneapolis, Minnesota.

Abstract

This paper presents a Network Publishing Paradigm (NPP) as a framework for building large scale Internet and Intranet (INET) systems. It changes the traditional function-centered information systems design paradigm by integrating content-centered design disciplines. NPP also categorizes the web site construction tasks into five different domains. We propose a layered methodology for web authoring where each domain is mapped onto a specific software layer. To maximize the utilization of layered software architecture, NPP modularizes technical elements and packages human expertise into interchangeable units that allow the construction of network publishing environments to be highly flexible and scaleable. These qualities, coupled with secure production protocols, are essential to commercial transactions on the INET. The NPP is a comprehensive development methodology that enables organizations to adopt late breaking technologies with minimal financial risks and managerial uncertainty.

1. Introduction

The explosive growth of Internet commerce has created a new breed of information industries with enormous potential. For instance, the Internet advertising industry amounted to $40 million in 1995, and is expected to be $312 million in 1996 (Based on Jupiter Marketing's estimation in 1996).

The authors have been working on commercial network publishing designed for Internet commerce for a couple of years. The issue of Internet commerce is different from setting up static

web sites (see table 1). For example, in Internet commerce, database support identifies the user and dynamically customizes both the content and the look and feel of the site. This database support enables large scale user load.

After much trial and error in developing commercial web applications, we propose a paradigm to address these issues from both technical and managerial perspectives.

Current technology does not provide automated solutions that integrate content assembly and user behavior analysis on a large scale. For example, when an online newspaper publisher wants to track the readership history of millions of subscribers, the publisher may analyze individual reader preferences, and dynamically assemble a customized newspaper [Kamba 96] for each reader. This task requires a major investment in customized software development. Even the simplest type of content assembly, such as reporting keyword search [Mendelzon 96] requires extensive programming labor. Researchers in areas such as intelligent agents and software robots [Williams 96] have worked on automated content search and reporting applications. However, the industry has not come up with a unified software architecture that captures and analyzes users' behavior and responds to it accordingly.

Web pages can be classified into 3 main types: 1) static; 2) semi-dynamic; and 3) dynamic.

1. Static Web Pages:
These web pages are created when a user writes HTML code either manually or with the help of tools such as Microsoft's Frontpage. These pages have static content and presentation once they are created.

2. Semi-dynamic Web Pages:
These can be further divided into two types:

a) High granularity level:
High granularity semi-dynamic web pages are those where the user creates a page template which is then parsed by a program such as Perl. The page displayed to the user can be changed by changing the page template. For example, Fusion from NetObjects provides support for high granularity semi-dynamic pages.

b) Low granularity level:
Low granularity semi-dynamic web pages are those where the user can generate templates for the components that make up the page such as widgets. These template components can then be reused for generating other page templates. For example, WebObjects by NeXT and Fusion by NetObjects provide support for low granularity semi-dynamic pages.

3. Dynamic Web Pages:
These web pages are made up of components that are objects whose type is determined at run-time (page-display) time. This provides the added flexibility that the pages can be changed based on user preference. For example, Manage!IT, VisualWave and WebObject provide support for dynamic web pages.

Table 1: Types of Web Pages

The current state of web authoring gives rise to several issues:

1. Web pages are replacing the roles of telemarketing personnel. This requires that the user interface of web pages be of comparable intelligence and attractiveness.
2. Consumer interest and individual user preference needs to be tracked to develop better marketing and sales strategies. This requires integrated information processing tools beyond traditional database technology.

Most Internet information providers still require programmers to manually create static (see table 1) web page (layout) templates. The attention of programmers is split between program functionality and content design. These web pages may not have the required persuasive value to attract and retain user attention because content design involves a different design aptitude than functional programming. Besides, information providers cannot dynamically assemble content (see table 1) for individual users, since user profile data is not associated with content data (see table 2). These deficiencies prevent effective content presentation, causing users to be constantly flooded with unnecessary information. With the current manual programming approach, developing Internet commerce applications continues to be an economically challenging business option.

Data Classification

A web-based information system contains data that can be viewed from three different perspectives. A well-designed web site has the ability to manage data in three separate categories: the content data, the presentation data and the user preferences data.

1. Content Data
Content data is the raw material of the information service. It contains the information that users want or need. The actual content can be textual data, images, sound clips, or other media formats. For example, in HTML, the actual content includes the content text, and the URLs to the media files.

2. Presentation Data
In order to present content data properly, we need to arrange the raw data in proper format and layout to increase comprehensibility. For example, in HTML, the format tags such as font size, table layout tags and line break tags are presentation data.

3. User Preference Data
A user friendly information system should capture user preference data as the third data category. That will help the system to deliver the most suitable content data and provide an automated mechanism to select the proper presentation format for the corresponding raw content data. This individualization should enhance the efficacy of the information system. One way to obtain user information is from the IP address of browser machine. The Internic database then can provide information about geographical location, business type, and connection speed of that particular machine.

Table 2: Data Classification

The lack of an Internet methodology for authoring has caused many companies to face three types of production problems: technological barriers, managerial problems, and the inability to keep information secure. Technologically, many INET information service providers lack a

highly reusable, flexible and timely architecture for the deployment of dynamic (see table 1) and creative content. Managerially, personnel from diverse backgrounds, such as content experts and software developers, have conflicting interests that can compromise the quality of the final products. In terms of security, most Internet information services do not have well thought-out plans for integrating security into every stage of their production process.

This paper proposes a methodology called Network Publishing Paradigm (NPP) in response to these challenges. NPP is a software production methodology specifically designed for INET application development. First, this methodology aims to shorten cycle times of software development by using a multi-layered framework, and second, this methodology establishes a new departmental organization; separating content generation, software engineering and systems support. These procedures ensure system integrity and information security. This arrangement resolves disparities in production paces across different departments. It also encourages departmental specialization and minimizes problems caused by the inherent departmental differences. Another implicit quality of INET information services is that the survival of the service depends on unique content and aesthetic designs that attract and retain user attention. To satisfy this new requirement, NPP introduces a new design paradigm. It moves the focus from function-centered design to content-centered design.

Traditional information systems only focus on supporting logical functionality and satisfying basic user needs. In contrast, designers of INET information services need to equally emphasize cognitive and aesthetic aspects of content as well as functional integrity. NPP integrates function-centered and content-centered design by creating a separate layer of software components that handles presentation and artistic content. This architecture retains the integrity of system functionality while allowing content experts and visual artists to attract and retain users' attention through creative and mutable designs.

NPP is based on experience derived from designing and developing large Internet commerce applications that minimize resource and effort consumption. NPP resolves the above problems by utilizing a five-tiered software architecture that:

- Separates content design from user interface programming.
- Provides data base support that allows users to change content and layout information. Thus, allowing not only pages that are individualized, but also allowing reuse of existing content, layout, and user-preference information in their creation.
- Includes an organizational structure that supports the development of such software systems, thus, maximizing collaboration among content experts and software developers.

2. Motivation

The World Wide Web (WWW) has had a dramatic impact on the design and implementation of information systems [Bernes 94]. Many corporations believe that delivering digitized information through the Internet and the Intranet (INET) is more versatile and cost effective than traditional networked database systems. While corporate information system designers benefit from the

portability of HTTP and simplicity of HTML, they now face many design and implementation issues. NPP addresses these issues.

- Prior to the WWW era, software engineers usually focused on the systematic management of a complex set of system functions. Content-oriented (see table 3) design is not the primary topic in software engineering, because it is domain specific. Each subject matter from a domain may have a different presentation style or a unique design philosophy. As traditional software engineering disciplines do not cover content-oriented design, web sites designed by software engineers may not present content effectively. On the other hand, content-oriented information experts, such as librarians, are not extensively trained in software engineering. If they are fully responsible for designing a web site, they may not know how to systematically manage complex computational services.

  Large commercial WWW projects need to manage both complex content presentation and sophisticated computational services. These projects need to merge both content-oriented and function-oriented design (see table 3) into one integrated methodology.

- Conventional web site development does not manage content data (table 2), user preference data (table 2) and layout data (table 2) in an integrated database. The content data is usually scattered across multiple files and directories. User preference data may or may not be integrated with the content database. Page layout data is hidden in HTML or other languages that can not be dynamically reconfigured at runtime. The isolation of these databases prevents web site developers from automating content customization for individual customers.

  Some web sites use programs that read HTML templates to automate content assembly. Each HTML template (table 1) contains a static page structure. The templates allow each page to be dynamically assembled with different content. However, the creation of each template is still labor intensive. The templates themselves are static, the format of each page cannot be changed by users at runtime. The template approach reduces the flexibility of content presentation. It also limits users' ability to setup their preferred page formats. (table 1).

1. Content-oriented Design

This focuses on the information domain and the way the information is presented to the user rather than the computational aspects of information presentation. Content-oriented design consists of three different aspects: a) content aspects; b) presentation aspects and c) user-preference aspects. The content aspect is concerned with the information on the page. For example, consider a page that displays stock information to a user. In this case, the content aspects would be the actual stock quotes and forecasts that are displayed. The presentation aspect is concerned with the layout and structuring of the content aspect. For instance, the stock quotes and forecasts can be displayed in the forms of tables or graphs. The user-preference aspect is the information the user is interested in and how the user wants it presented. For example, a user in London may be interested only in British stock quotes but not in American quotes and forecasts.

2. Function-oriented Design

This focuses on the business application, business logic, business rules and computational aspects of the information rather than the presentation of information to the user. For example, the function-oriented part of a page that presents stock information to the user will focus on forecasting algorithms, network optimization so that the information can be updated in a timely manner and other computational aspects rather than the graphic design and layout.

Table 3: Content-oriented vs. Function-oriented Design

- Many web sites allow application developers and content authors to work on a centralized developer server directly. This process usually induces conflicts when many people are updating their source code or content simultaneously. It may also generate redundant user interface programming work. For instance, after content authors create marked up user interfaces in HTML (or other languages), developers need to convert these HTML files into template files for dynamic content generation. This process dependency creates multiple phases of redesign and verification, which is time consuming and expensive.

- From a management point of view, web sites are content-centered, unlike traditional function-centered software systems. Sometimes, programmers are given the authority to write and design content presentation or page layout. Even with advanced tools, programmers may not have the skills to design the organization of the documents. Having programmers create the final version of the page layout may compromise the original design intention of domain experts. We need a specific production management methodology to maximize programming efficacy and to resolve these differences in artistic talents and technical aptitude.

3. Network Publishing Paradigm

This section proposes a software architecture along with an organizational structure to tackle the problems outlined in section 2. We propose a layered methodology for web authoring and publishing. In section 4, we provide an example walk-through of the steps to be followed in web authoring using NPP. Readers interested in the steps to be followed rather than the details of the individual stages can read section 4 first.

The layered approach to problem solving is well known and has several advantages such as decoupling, localization of dependencies, support for standardization and potential for reuse [Buschmann 96]. However, the issue that needs to be addressed is allocation of function to individual layers so that these advantages can be realized.

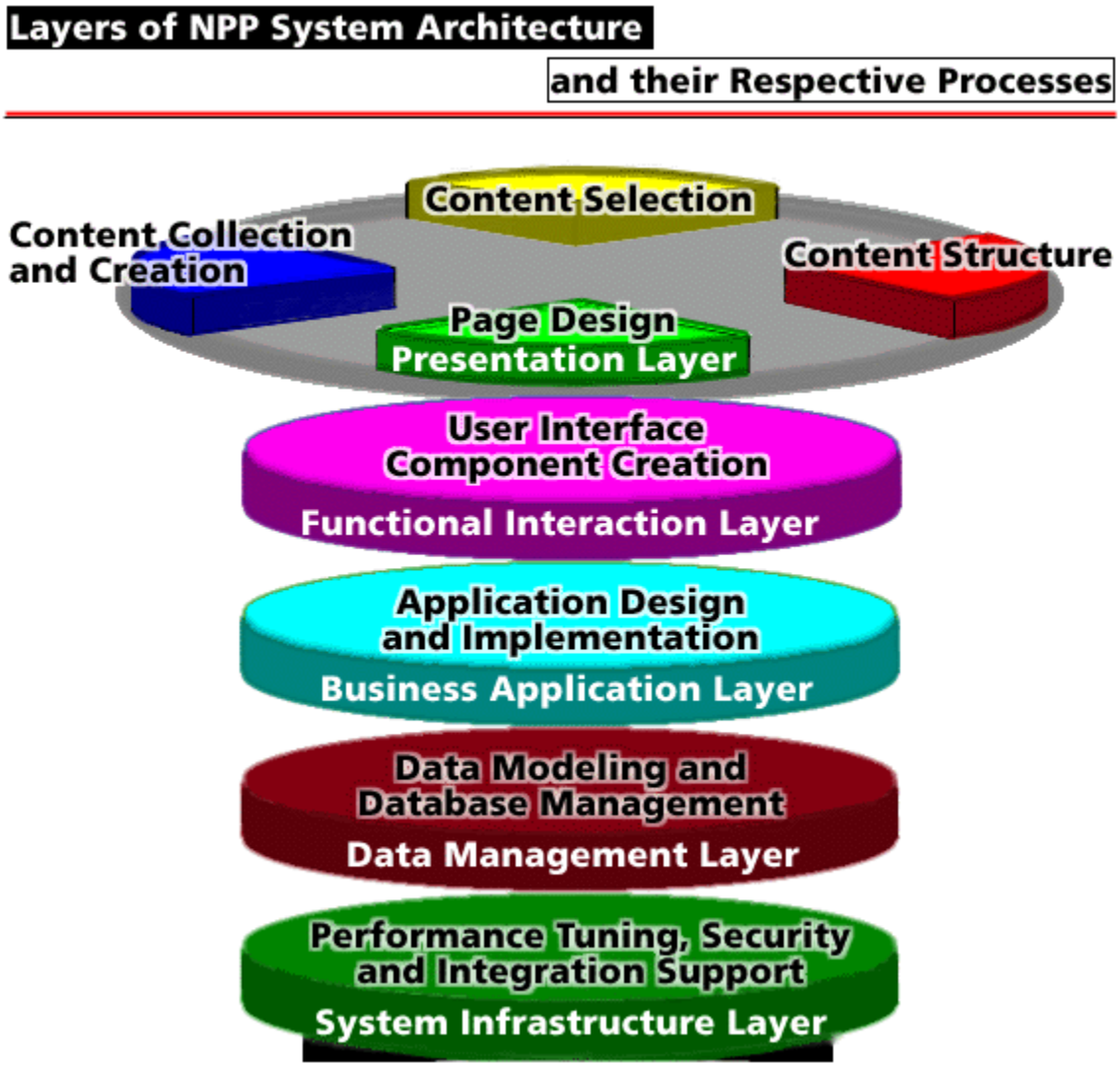

Figure 1: Network Publishing Paradigm

We found that not only can the software architecture be partitioned to realize these goals, but also the knowledge that goes into web authoring can be layered. NPP presents a layered approach in which both the knowledge domain and the software architecture are layered. Broadly, the goals of NPP are:

1. A layer only uses services that are provided by the layer below it;

2. Each layer has a distinct functionality and knowledge domain; and

3. People can be allocated to each layer in a manner that minimizes the inherent conflicts that arise when people of different backgrounds work in a common environment (see section 4).

Table 4 describes the services used by each layer, the functionality of each layer, the expertise required and the participants that provide their expertise to each layer.

Layer	**Service used**	**Functionality**	**Expertise required**	**Participants**
Presentation Layer	Functional Interaction Layer Services	Content selection, creation, collection, structure management and page design	Content subject, ergonomics	Domain experts, Media artists, Information architects.
Functional Interaction Layer	Business Application Layer Services	User interface component selection, creation and management	Content subject for page design and UI selection, Software engineering for UI creation and management	Layout artists and Information architects for UI selection and page design, software engineers for UI creation and management.
Business Application Layer	Data Management Layer Services	Business application design and implementation	Business rules and applications	Domain experts, application programmers, application system architects.
Data Management Layer	System Infrastructure Services	Database support for content, layout and preference databases and application data modeling	Database and DBMS concepts	Database administrators and modelers
System Infrastructure Layer	Networks, Operating Systems et cetera.	Performance, security and integration	Software and hardware maintenance, security, networks, operating systems et cetera.	System administrators

Table 4: NPP Layer Information

Publishing content and computational services on the INET involves the interaction and knowledge of several people. The proper mix of talent should include content experts, software developers, database administrators, and network engineers. Traditional software methodologies do not discuss how to integrate workers from diverse backgrounds to work in a common network publishing environment. To address this issue, we categorize the publishing tasks into five layers. These layers are partitioned so that not only the software architecture is layered, but also the knowledge required of the participants in each layer is kept as independent of other layers as possible. This specialization minimizes the interpersonal friction that often compromises the quality and quantity of work. In this section we describe NPP as it is used for web authoring. A layered architecture with standardized interfaces between the different layers makes the application development task easier. Creating a general application now just requires the assembly of components from different layers.

3.1. Presentation Layer

Knowledge Required

This layer requires expertise in the content subject that is usually unrelated to software engineering and computer technologies. For example, if the INET information service is about income taxes, we need accountants that are specialized in tax laws. This domain also involves professionals who specialize in content grouping, indexing and visual presentation. Such knowledge is usually related to library or cognitive sciences.

Processes

1. *Content Selection, Creation and Collection:*

Content selection defines the scope of the information based on user-preference studies. The creation of content elements involves content experts and media artists working together. Content experts write textual descriptions and select visual aids that illustrate the actual content. Media artists, which include grammaticians and digital media artists, help edit the content and provide artistic expertise to enhance the content presentation. In some cases, audio and animation is also part of the content requirement. Media artists should help content experts to produce those media files.

Summary 1:

- **Participants**: Content Experts and Media Artists
- **Tools**: Text editors, image editors and other media authoring tools. Examples of such tools are Microsoft Word, FrameMaker, Adobe Photoshop and Illustrator. Many content artists also need animation and sound editing tools.
- **Examples:** For an online magazine about movies, content experts may include movie critics and movie directors, where media artists include graphics artists, audio digitizing experts and animation creators.

♦

2. *Structure Creation:*

To present content in a cohesive and comprehensive manner, it is critical to have a smooth content flow that associates different pages. However, when a web site involves many pages and

many different content subjects, the design of content structure becomes difficult. Information architects are people who structure the content on the web pages for greatest user impact.

Summary 2:

- **Participants:** Content Experts and Information Architects.
- **Tools:** Visual or character-based tools that help information architects to create hyperlinks and establish document relationships. These structures and hyperlinks help readers navigate across multiple pages or documents.
- **Example:** Fusion from NetObjects includes a site structure editor that allows information architects to create page relations graphically.

♦

3. *Page Design*

Page design requires expertise and tools different from those of content creation. Content experts should arrange content elements into different pages based on content relevance, comprehensibility and presentation efficiency. Page layout tools can be used to manage the locations of visual and textual content elements in each web page. To have a consistent style on a web site and increase aesthetic qualities of each page, we need inputs from people who specialize in page layout and visual information presentation (Page Stylists).

Summary 3:

- **Participants:** Content Experts, Page Stylists.
- **Tools:** Visual or textual page component layout tools that specify the locations of components on a page or a window.
- **Example:** Examples of tools include WebObjects Builder from NeXT, FrontPage from Microsoft, Fusion from NetObjects.

♦

The information presentation layer contains software modules that utilize objects extracted from the functional interaction layer and organize them to present data to users. Software modules in this layer control the layout, content and aesthetic aspects of a page. A majority of these modules should be stored in the database for dynamic layout generation and accessed using database associations present in the other layers.

This layer is created to allow content experts to locate and replace user interface components for better content presentation. Therefore, it is preferable that the information presentation layer does not include complex user behavior logic and data retrieval code. The components in this layer specialize in component layout logic, aesthetic disciplines and session management. The content experts can organize components in this layer to create reusable design styles and media presentation templates. Reusing components in this layer can reduce a large portion of layout design and layout programming labor.

Software components in this layer combine UI components to present content data and retrieve user submitted information. The primary function of this layer is to effectively present and manage information from a user's perspective. An information manager should have three view points to manage the components in the this layer: 1) window/page relationships (information

structure), 2) component layout in a window (information presentation), and 3) media files (aesthetic style).

3.2. Functional Interaction Layer

Knowledge Required

Unlike the previous layer, this knowledge domain is more related to software engineering and is less specific to content subjects. The design and implementation of reusable user interface components should be applicable across different content subjects. For example, a text field in an HTML form allows users to submit a string back to the server. The interaction functions in a text field, such as keyboard event processing, and simple input validations are not specific to a specific content subject. However, this domain still needs content experts to provide design guidelines. The Chief System Architects need to understand what type of user interface function should be provided for a given content subject. The User Interface Programmers should work with Chief System Architects to determine how to structure the code base to maximize its reusability.

Summary 4:

- **Participants:** Chief System Architects**,** Content Experts and User Interface Programmers.
- **Tools:** Programming Languages and corresponding compilers or interpreters. They may be Java, JavaScript, HTML or Microsoft Visual Basic Script.
- **Example:** Text fields, buttons and drop down menus, drag and drop capabilities, information visualization widgets, input validation algorithms, Microsoft user wizards.

♦

Processes

A UI Component is an individual object in a page, such as a text field or a button. The positioning of these objects is managed by software modules in the presentation layer. This concept of separating UI Component from other software layers is desired for content-oriented design. It differs from the traditional Microsoft Visual C layered approach, which combines UI components with UI layout. By independently managing layout logic and component level logic, NPP allows content experts to manage the layout logic by selecting and placing complex components programmed by software developers. This approach speeds up application development. It also indirectly increases the quality of content presentation.

UI components may contain extensive user behavior logic that validates data input, monitors mouse movement and warns users about illegal user actions. By detaching UI components from domain specific business logic, UI components can be applied across different projects and business domains. The primary purpose of building components in this layer is reusability.

The UI layer contains reusable objects [Mowbray 96] that display data in a page. They may be simple buttons, text fields or sliders. They can also have sophisticated logic built into the components. The objects in this layer must fulfill the following requirements:

1) UI components should not be domain specific. Components in this layer should be generic, so they can be reused across different projects. However, there are always content specific UI components that are highly correlated with a project domain. For example, a travel agency web

site may employ intelligent user interface agents that help novice users explore interactive maps. Even these content specific user interface components may still have common code segments that can be reused [Poonawala 96].

2) Should maximize client side logic when possible. Having intelligence on the client side will help users to realize their mistakes quickly. JavaScript and Java [Java 96] can be used to embed logic in each user interface component.

3) Should implement the UI of production tools in HTML or Java. This is mainly an economic issue. Building user interface in HTML and Java provides cross platform compatibility. In most cases, HTML can be used to construct simple form based interfaces faster and easier than any other programming languages. Implementing tools that are accessible over the network can allow administrators to solve problems and carry out production tasks remotely.

3.3. Business Application Layer

Knowledge Required

Most INET information services have specialized computational tasks that need customized software programs, or business applications. The implementation of business applications involves software engineering expertise and content specific knowledge. The Content Experts should provide the business logic and functional specification of the applications. Then, each application is assigned to an Application System Architect who designs the application with Content Experts and programmers.

Summary 5:

- **Participants:** Application System Architects**,** Content Experts and programmers.
- **Tools:** Programming Languages and corresponding compilers or interpreters. They may be C, C++, Perl, Java, or other programming languages that are suitable for business logic programming.
- **Example:** Home banking, network computer games.

♦

Processes

Business logic components are objects that implement business logic and business rule algorithms. Business objects communicate with other business objects to acquire information and computation results.

An information service application has two areas of business services, a) user services and b) administration services. User services refer to programs that are used by the end users. For example, key word search is a computational service provided to users. Administration services refer to programs that are used by server administrators or content providers. For example, user statistics analysis is a service mainly used by content providers.

Some business logic components can be as simple as a data fetching agent, which associates data with user interface (UI) components. It may also have complex algorithms that carry out the business logic.

3.4. Data Management Layer

Knowledge Required

This knowledge domain specializes in data storage, retrieval and integrity. To manage data for a particular application, we need the Application System Architects to design a functional data model based on information captured from Content Experts. Content Experts are included in this domain only for concept verification. Application System Architects need to work with Database Administrators to choose proper database engines depending on application and content needs.

Summary 6:

- **Participants:** Application System Architects**,** Database Administrators, Database programmers and Content Experts.
- **Tools:** Programming Languages that work with specific Database Engines. The programming languages may be SQL, 4GL.
- **Example:** Reuters news database, Lexis-Nexis professional electronic document database, Yahoo.

♦

Processes

The data management layer contains the data access logic and the structure of data storage. It allows data to be shared between users and applications and also provides data security, extensibility, data distribution and integrity.

In order to provide the content dynamism required for automatic content production and assembly, we need to fully utilize database technologies. In the web publishing industry, the content subject can be roughly divided into three categories. They are content data, user preference data, and page layout data (data regarding the typesetting information) (see section 3). Therefore, the central database is partitioned into: 1) content database, 2) user preference information database and 3) page layout database. Managing these data bases in a centralized database provides the association between content, layout and user preference data. It enables the system designer to customize content and layout based on user preference. Layout designers can also store and retrieve their experience thus reducing design effort by reusing pre-built templates.

The data management layer takes care of the following issues:

1) Stores layout data in the database. To dynamically assemble content based on each user's preference, the layout preference information needs to be stored in a database. Currently, page layout information is hidden in HTML code or other programming languages. To present content in a different page layout format, programmers need to manually create a new layout template. By storing layout data in a database, the page format can be changed dynamically based on the layout data given by the database each time.

2) Stores user preference data in the database. Many web sites have started storing user preference data in the database. To maximize user interest, it is necessary to associate user preference data with content and personal layout preference.

3) Stores content structure in the database. Content data can also be stored in the database. Considerable work has been done in the area of database requirements to store content data such as images and audio [Wu 95].

4) Provides a uniform mechanism for accessing different type of information. Given the three types of data, some of them may be stored in different databases or different file systems. It is necessary to have a unified data access layer to access and integrate all these databases. This will provide tremendous business value to content assembly and management. In many programming environments, the language vendors provide a set of database adapters [NeXT 96] to different database engines.

By associating user preference data with content data, we are able to customize content based on users' site visit history. By storing layout information on the database, we enable users to customize their pages to best suit their needs. This is important for building a dynamic information service system.

Using a separate layer to provide a unified data storage and retrieval provides better data security extensibility, distribution and integrity.

3.5. System Infrastructure Layer

Knowledge Required

This knowledge domain involves the installation, and maintenance of the software and hardware infrastructure that supports the entire production environment. This includes installation of third party software applications, document servers (such as HTTP, IIOP), database engines, operating systems and other hardware andsoftware support devices. It also covers standard security services and data backup services. System Administrators are responsible for the entire System Infrastructure. They need to understand the current and future needs of the entire organization in terms of Operating Systems, Database Engines and hardware performance.

Summary 7:

- **Participants:** System Administrators**,** Database Administrators.
- **Tools:** System scripting languages, data backup devices, Software and Hardware tools for performance monitoring and security protection.
- **Example:** Choice of hardware, software combination. network topology design, system security support.

Thus, we see that the five distinct domains can be used to distinguish and partition the application development effort.

Processes

The system support layers consist of the hardware and software infrastructure. The components in this layer include the HTTP server software, operating systems, database engines, network protocols, hardware platforms and other hardware accessories. The selection of software and hardware systems will impact the performance, portability, scalability, and stability of the entire production organization. Separating these elements from other layers reduces the dependency on particular software or hardware vendors.

The inclusion of HTTP services differentiates web publishing system support from traditional database system support. Unlike Client/Server or mainframe databases, HTTP introduces the concept of session management and state management. The software programs developed to run under HTTP service need to track state or session variables explicitly. The best solution is to choose a commercial web software development environment that supports session and state management.

The system support layer should take care of three issues: a) *system performance*: It should provide the choice of hardware and software combination that satisfies production people's needs. That includes network transfer rates, compiler speed, database transaction rates and general computation tasks; b) *system security*: the traditional security devices include physical security, encryption algorithms, password management and remote access management. For web publishing services, many security systems are integrated with session management. It is critical to prevent security breaches that come from insecure session management, and; c) *system integration*: it should provide the ability to integrate computing platforms that have different operating systems, network protocols and hardware components. It should also allow sharing of data between computers and network file systems. Integration with existing legacy systems is also required as many corporations would need extensive legacy system support to utilize old data.

The advantages of defining a separate infrastructure layer to support the entire production is that it eliminates redundant security and backup tasks from other layers. It specializes in system performance and network stability issues. When this layer is an independent entity, layers built on top of it will also be designed to accommodate the possible change of the operating systems, network protocols, *et cetera*. This will reduce the production environment's dependency on particular vendors' products. It also makes integration with legacy systems easier.

4. A Walk-through of NPP

Figure 2 depicts the process followed for web authoring using NPP and the layers in which these steps fall. The grayed boxes depict the NPP process. We provide a walk-through of the process followed for web authoring using NPP by means of an example.

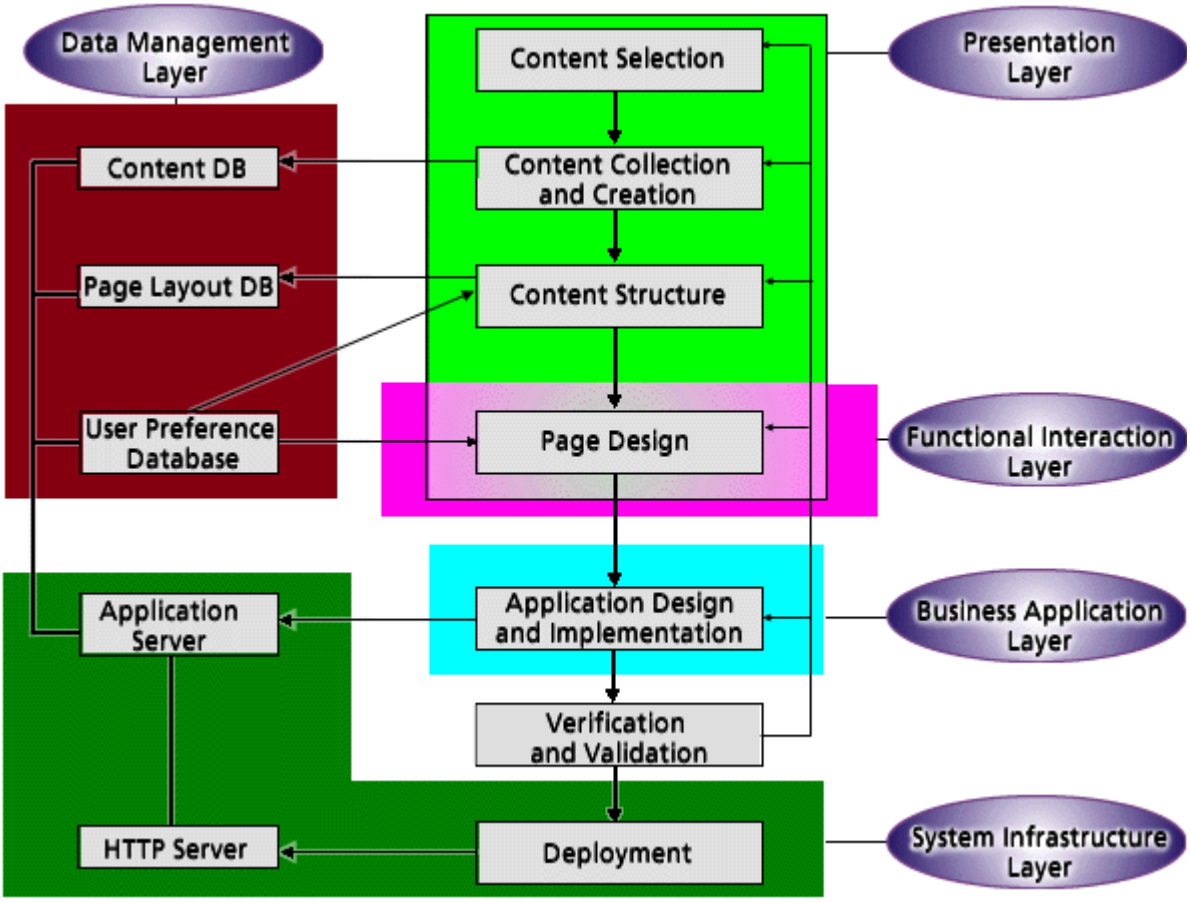

Figure 2: NPP Walk-through

Example:

To illustrate the NPP processes in each layer we use an Internet banking service as an example. ABC Consumer Bank wants to offer Internet online banking services. They found that using the Internet offers their customers more personal services and reduces check processing costs. After watching other banks offering Internet banking services for a year, ABC found that many of their current customers are transferring funds to their competitors. The Board of Directors decide to offer competing services on the Internet. Before ABC actually offers full consumer banking service on the Internet, it wants to have a web site that promotes its upcoming Internet banking services and provides trial memberships to selected customers to test the online banking operation. However, the existing information system staff does not have experience in providing web-based information services. The existing mainframe and client-server development workflow take two to three years before reaching the deployment stage. After doing some research, they decide to use the NPP and followed its processes.

1. Content Selection

ABC first needs to figure out what type of services they want to provide, and which associated legal and promotional documents that they need to present to potential and existing customers. To select proper content for Internet publication and advertisement, they need to involve

financial specialists and market research experts. These domain experts will conduct multiple brainstorming sessions, investigate related government regulations, analyze market needs and look at existing Internet online banking services. These activities will generate a number of feasibility reports and the requirement documents. These documents will need to clearly define the scope of each Internet banking service and corresponding promotional material.

2. Content Collection and Creation

After the content selection process, the content topic and intended audience are defined. The bank puts together a development team to collect the marketing material and service policy documents that already exist within the organization. In cases where the new services differ from traditional banking services, the internal banking experts need to work with technical writers, and lawyers to create related documents.

An information architect should work with a data modeler to design a unified content data model that captures the relationships between content data elements. In this project, content data are classified into the following categories:

1) Account Policy Documents - These types of documents are straight text files that illustrate the legal agreements between the bank and its customers. The documents are created by lawyers and paralegal writers.

2) Promotional Information - This includes company mission, history, and customer satisfaction reports. This information is straight text data that can be recorded as long text data in database engines such as Oracle.

3) Operational Information - This information type includes customer service telephone numbers, email and contact persons' names. These data items can be captured with traditional relational database modeling techniques.

4) Corporate Identity Information - The corporate logo, standard color palettes and other artistic images that are independent from the data structure of the previous three information categories.

To increase user friendliness, user preference data such as personal survey answers, preferred service types (personal banking and small business banking), scope of account history, personalized stock ticker selections and other personalized economic reports are needed. When such user preference data are provided, they should be integrated in the content data model. All this information can be captured using logical data structure modeling techniques.

In the mean time, media (graphic) artists should design the site style and create standard color palettes for presenting a consistent corporate image throughout the site. The textual information, corporate logos, color palettes and media files are managed as raw content data.

After the data model is programmed into the database engine, database administrators work with information architects to load all data elements into the content database engine. Later, the System support department should supply standard load scripts to reduce manual data loading/updating labor. The Content Creation and Collection processes involve creating individual data elements, loading all data elements into the central content database, and continuously feeding new data elements into the content database.

3. Content Structure Design

To effectively lure potential customers to read through the promotional information and sign up for trial banking membership, the content flow should be cohesive and easy to follow. Information architects, marketing specialists and banking experts should work together to have a clear content outline and to precisely define the hyperlinks between pages. The content outline illustrates the main topic as well as the exact data elements and computational services contained in each page. (For the trial online banking services, these pages would need to calculate quarterly interests, account summary, and fund transfer.) In this process, the information architects may use tools similar to the Site-Structurer of Fusion (NetObjects) to build a mocked-up site to validate the content structure design.

4. Page Design

Given the site structure and content outline created in the previous process. The page stylists and graphics artists can start modifying and designing individual pages. Aesthetically pleasing look and feel helps the web site to capture Internet users' attention. The artistic quality of the web site improves market response drastically. A consistent artistic style across the entire web site demonstrates corporate commitment to the Internet banking service. It is the critical factor to boost customers' confidence level and triggers their desire to try out the online banking service.

While the page stylists and graphics artists are beautifying pages, the Information architects also need to verify that content presentation is not obscured by flashy graphics. Information architects should also help media artists to reduce the size of image files transferred over the network. By using consistent corporate icons, standard color palettes and image pallets, browsers will automatically reuse the cached media files. The caching of files decreases page response time and reduces server load and network traffic.

Given the bank’s experience in designing user interfaces for Automatic Teller Machines (ATMs), the pages for actual banking services are rather simple and robust. Page designers simply need to touch up the look and feel of the page.

5. Application Design and Implementation

ABC has a large internal staff that has extensive client-server banking application development experience. They found that since the user interface on the web browser is almost identical to the standard ATM’s, they can reuse a large part of their banking software framework. The Application System Architect specializes in client-server computing, and he first thought that the project would be a trivial task.

However, the System Architect along with his staff found one unfamiliar programming task that drastically differs from their mainframe and client-server development experience. The standard HTTP server does not keep track of the states between pages. It also cannot tell the difference between multiple browsers on a machine. These two problems introduce extra programming labor and create a potential security issue. The technical people explained this problem to the management, and stated that it may take more time to develop the web-based application than the traditional client-server or mainframe applications.

The management started out by attending Internet Commercial Application seminars. In one seminar, they were told that there are vendors who provide development tools to track states between pages and manage sessions for multiple browsers on a machine. (WebObjects from

NeXT, Visual Wave from ParcPlace are two such commercial tools available today). So they contracted information technology consultants who are familiar with these tools to demonstrate the technologies and evaluate their performance and functionality.

The evaluation came back with very positive results. Development tool vendors provide well-designed programming interfaces and state/session management frameworks that almost completely eliminate the need for an application programmer to explicitly track state and session in application code. They compared these vendors and selected a development tool vendor as the standard development platform.

The Application System Architect could now start transcribing the existing client-server data model and object model to fit the web application. They found that the data and object model needed little modification. They proceed with programmer training and start implementation when programmers are conversant with the new development environment.

6. Verification and Validation

After finishing the prototype, they find that testing, verifying and validating the web-based system is similar to their traditional testing procedures. They generate bug reports and continue on with bug correction. After a few iterations, the application is validated and authorized for production.

7. Deployment

Before deploying the entire web application along with promotional material on the Internet, ABC expected three hundred thousand visits per day. They found that the average work station and T1 connection would not be able to sustain the load. The System Administrators who specialize in performance tuning and network optimization plan out the required computer resource and network bandwidth. They again verify and validate the performance result before deployment. They also test and validate an extensive data security and password management plan. Finally, they deliver the content and basic banking services to selected customers.

5. NPP Organizational Structure

In this section we describe the departmental structure required to support NPP. We propose three separate departments, each in charge of one or more of the layers of the production methodology. These departments are: 1) the media production department (section 5.1); 2) the software engineering department (section 5.2), and; 3) the system support department (section 5.3). The advantage of this breakup is that each department is in charge of a different aspect of web authoring and therefore achieves division of labor and efficient deployment and utilization of resources. For example, the media production department controls the presentation of information, builds non-reusable software components, such as user specific layouts internally and uses reusable components developed and documented by the software engineering department. This arrangement will avoid the content deadlines that can affect the quality of software produced by software engineering department. At the same time, this layered organization enables content experts in the media production department to rapidly prototype time critical functionality while having the engineers in software engineering and system support departments continuously improve the performance and flexibility of the overall system.

5.1. The Media Production Department

The Media Production Department is content-oriented, and its production schedule is based on the nature of content. In many cases, contents are highly time sensitive (news or stock price reports) and its production schedule should be controlled by people who are content-aware.

The people in this department include the following:

1) This department should be headed by a production director who is a content expert, controlling the quality of the entire site. This person will direct a number of content experts and information architects who are in charge of individual projects. The production director, hires and manages the content artists, which may include writers, visual artists, musicians and animators. The production director communicates with the marketing department or other divisions in the enterprise that provide strategic input for the economic and social values of the content. The production director can make decisions on how to prioritize the resources for the content experts. The production director is also the mediator between information architects and content experts.

2) The content experts who are in charge of individual projects. Content experts should consult the production director for priority production resources.

3) The information architects who work with the content experts to design the structure of the documents. When all pages are finished, the information architects work with the quality control department to test the page. They look for and correct errors. They then release data to the external site.

4) The layout architects who work with the information architects for page level information presentation

5) The media artists who continue to prepare new images or animation clips. The artists also directly receive instructions from information architects for content specific projects. The images or animation files will be put together by layout designer.

6) Application system architects and application developers when a project needs complex application development such as an Internet banking application. There will also be a few application programmers that can assemble an application based on the framework provided by the software department. Each programming project is assigned to an application system architect who will work with the information and layout architects for technical support and feedback. The application developers are the interface to the software engineering department. They will learn to use new software libraries developed in the software engineering department and give suggestions for change.

To create compelling content, this department needs to promote creativity. To deliver time critical information such as stock quotes, this department needs to control work flow efficiently. There, this department controls the content. It is directly responsible for the deployment of content and software. It also produces images, audio files and animation that are required for content presentation. It should have full time programmers that prototype dynamic content generation for short term projects. The department is in charge of the entire presentation layer. In general, this is the “consumer” of the other departments’ products and services.

5.2. The Software Engineering Department

The software engineering department is technology-oriented. Its goal is to provide highly robust technology solutions and reduce production cycle time by increasing flexibility and reusability. This department produces and manages reusable components such as UI components and business objects. It builds the software framework for the entire organization. The people in this department include:

1) The framework system architect who is in charge of the department. The framework architect hires and manages the direction of the department. The system architect should work with programmers, technical writers and test engineers to document the specification, design, and test cases.

2) UI programmers, business logic programmers and database programmers who are assigned work on the appropriate layer of the software tool set. Thus the department is conceptually divided into three divisions, the UI division, the business logic division, and the database division. All programmers should be cross trained to perform tasks in any department, but they should also emphasize their expertise in one division. Especially for large projects, it is imperative to have people working on multiple layers of the software simultaneously. The system architect should understand all layers of the system and help design the programming interface between layers.

3) The test engineers who design test cases and generate test data when feasible, and test the components after implementation. They are responsible for publishing the components as a reusable component, ready for use and user comment.

4) The technical writers who are responsible for the documentation. Good documentation and modularized test cases also increase the traceability and reusability of each software component. Unless the components are well documented, it would be difficult to let any other programmer share code base and reuse components.

5.3. The System Support Department

The system support department is resource-oriented. It supports the entire organization's computational infrastructure. This department manages network and computer infrastructure. The main purpose is to maintain the performance and stability of the computer network. In terms of the layered architecture, this department handles the system support Layer. In terms of management responsibility, it plans the required network resources and computer performance along with the projected demand coming from all departments. In a web publishing environment, this department has the extra responsibility of security related tasks. It needs to monitor user log files, prevent external intruders and monitor internal illegal data access.

There are three divisions of tasks under system support department: 1) maintenance, 2) security and 3) resource administration. The three divisions are headed by a system administrator, who oversees the entire operation and communicates with other departments for resource allocation.

Many network engineers and maintenance people are also involved in this department to ensure the operation of the computing infrastructure.

6. Advantages and Applications

Some of the advantages that the NPP methodology described in sections 4 and 5 provide to web authoring and publishing are summarized below:

- A layered architecture provides maximum application flexibility and minimizes vendor dependency and insulates programming complexity by separating UI components programming from UI layout code (UI Component layer vs. Presentation Layer)
- Storing layout information in database can eliminate layout programming labor and enable reuse of layout design from the database.
- NPP provides a standardized Application Programming Interface from UI components to Business Logic. Thus, most applications would only need to implement business logic. All the UI component and UI layout code can be reused from the database.
- Applications written in standard languages that can be portable over multiple platforms increase the ability to partition applications across multiple machines for performance enhancement.
- Access to the database is through a standardized interface (database adapter layer). This reduces design dependency on a particular database vendor or database technology.
- Provide session security and revision control.

Some of the applications that would benefit from the NPP methodology for web authoring and publishing are:

- Electronic newspaper publishers and content providers who supply individualized information services to each subscriber.
- Information dependent professionals (Stock traders) who need to have effective user interface to a dynamic content database.
- Advertising agencies that want to target individual web users by customizing advertising content and presentation style for each user.
- Individual users who want to configure their own content database for effective information acquisition.
- Organizations who want to reveal their corporate documents to different users at different levels and different content interests.

7. Comparison with Existing Work and Future Directions

There are many products out on the market that utilize content-oriented design methodologies. This section describes some of them and how they compare to the NPP methodology.

1) Fusion by NetObjects, Inc. - Fusion is an integrated web publishing software product. This product clearly separates the publishing process into five separate phases, based on [Mok 96]. Each uses a different set of tools for content management and content authoring. However, this product does not have extensive database support, and lacks mechanisms for revision control. Fusion manages pages by generating a number of HTML files. The page layout information is hidden in HTML. Its site structure is hidden in a proprietary binary format. This hinders future integration and expansion.

2) Manage!IT by Imperative, Inc. - Manage!IT is one of the most comprehensive web publishing software product on the market. It has extensive database support, and utilizes an object-oriented data model for page formatting and content assembly. It integrates revision control systems and access control devices. It also utilizes a three-tiered architecture for software reusability and system flexibility. This product has the most complete feature set in Network Publishing. However, its three-tiered architecture combines information presentation with the user interface components. This would force content experts to work directly with user interface programmers. Thus, creating extra communication overhead. Besides, companies employing a powerful product still need a well-defined production methodology that is compatible with both content publishing and software development.

The HTML based WWW is not the ultimate network publishing standard. Many new network protocols have better data transmission efficiency and offer larger extent of flexibility. Newly introduced technologies, such as Java and its distributed computing model have set precedents in Internet commerce.

The NPP method is not constrained to the development of WWW based systems. It can also be applied to content-intensive information systems. The technology of storing information presentation styles in a database is essential for high volume content publishing environments.

The NPP method expands on the concept of information classification and code reuse first popularized by Object-Oriented techniques. Object-Oriented design encourages software designers to map the problem domains into well-defined “classes”. Each “class” will selectively publicize its internal state information and functions through a public interface. This classification approach helps reduce the size of each problem domain. The publicized interface provides a standard protocol that enables each “classified” object to communicate with others. Thus, all “classes” can concentrate on their own work while sharing the expertise in other classes.

The NPP classifies applications into five functional domains. Each domain publicizes standard programming interfaces for inter-layer communication. It helps each layer to specialize in its domain functions, and share results with other layers. The departments are also classified based on the nature of task schedules and production expertise. Thus, each department can specialize in its tasks and reduce redundant work.

8. Conclusion

The lack of an INET methodology for authoring has caused many companies to face three types of production problems: technological barriers, managerial problems, and the inability to secure information. Technologically, many INET information service providers lack a highly reusable, flexible and timely architecture for the deployment of dynamic and creative content. Managerially, personnel from diverse backgrounds, such as content experts and software developers, have conflicting interests that can compromise the quality of the final products. In terms of security, most Internet information services do not have well thought-out plans for integrating security into every stage of their production process.

This paper proposed a Network Publishing Paradigm (NPP) in response to these challenges. The NPP is a software production methodology specifically designed for INET application

development. First, this methodology aims to shorten cycle times of software development by using a multi-layered framework, and second, this methodology establishes a new departmental organization; separating content generation, software engineering and systems support. These procedures ensure system integrity and information security. This arrangement resolves disparities in production paces across different departments. It also encourages departmental specialization and tolerates the necessary differences in point of view between them.

Another implicit quality of INET information services is that the survival of the service depends on unique content and aesthetic designs that attract and retain user attention. To satisfy this new requirement, NPP introduces a new design paradigm. It moves the focus from function-centered design to content-centered design. Traditional information systems only focus on supporting logical functionality and satisfying basic user needs. In contrast, designers of INET information services need to equally emphasize cognitive and aesthetic aspects of content as well as functional integrity. NPP integrates function-centered and content-centered design by creating a separate layer of software components that handles presentation and artistic content. This architecture retains the integrity of system functionality while allowing content experts and visual artists to attract and retain users' attention through creative and mutable designs.

As, Richard Gabriel [Gabriel 96], points out,

> *"Habitability is the characteristic of source code that enables programmers, coders, bug fixers, and people coming to the code later in its life to understand its construction and intentions, and to change it comfortably and confidently."*

In order to obtain better "habitability" for Internet application development and deployment, we will need to orchestrate the organizational structure, software tools and operating procedures in a unified theme. The Network Publishing Paradigm introduces the foundation that will increase the comfort level and production efficiency for Internet application development and web authoring.

Acknowledgments

We would like to thank Mr. Mark Berman of NeXT Software, Inc., David Appleby, Stephanie Delzel, Randy Marchessault, Steve Martin, Cynthia Schuster, and Odin Soli of Integrity Solutions, Inc., Eagan, and the graduate students at the University of Minnesota, Minneapolis for their contribution in developing and implementing this framework. We also thank Carl Steadman, the Production Director of Hotwired, San Francisco, for validating this paradigm.

References

[Bernes 94] T. Bernes Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The World Wide Web. Communications of the ACM, pp 76-82, August 1994.

[Buschmann 96] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. A System of Patterns. Wiley Publications, 1996.

[Gabriel 96] R. Gabriel. Patterns of Software - Tales from the Software Community. Oxford University Press, 1996.

[Java 96] Sun Microsystems, *The Java Language Tutorial*, JDK version 1.1, http://java.sun.com.

[Kamba 96] T. Kamba, K. Bharat, and M. Albers. The Krakatoa Chronicle - An Interactive Personalized Newspaper on the Web. Fourth International World Wide Web Conference Proceedings, 1996.

[Mendelzon 96] A. Mendelzon, G. Mihaila, and T. Milo Querying the World Wide Web. Technical Report, University of Toronto, 1996.

[Mok 96] C. Mok. Designing Business. Hayden Publications, 1996.

[Mowbray 96] T. Mowbray and R. Zahavi, *The Essential Corba: System Integration Using Distributed Objects*, Wiley Publications, 1995.

[NeXT 96] NeXT Software, Inc. Enterprise Object Framework Developer's Guide. Addison Wesley, 1996.

[Poonawala 96] M. Poonawala, S. Subramanian, W. T. Tsai, R. Vishnuvajjala, R. Mojdehbakhsh, and L. Elliot. Testing Safety Critical Systems - A Reuse Oriented Approach. Technical Report, University of Minnesota, 1996.

[Williams 96] J. Williams. Bots and other Internet Beasties, Sams Publishing 1996.

[Wu 95] J. Wu, A. D. Narasimhalu, B. Mehtre, C. Lam, and Y. Gao. CORE: a Content Based Retrieval Engine for Multimedia Information Systems. Multimedia Systems, pp 25-41, 1995.

Credits

Editing, graphics makeover, and layout of *Network Publishing Paradigm* were furnished by the International Interactive Communications Society - Minnesota Chapter, Jerry Schlei, Webmaster.

In cooperation with principal author, Benjamin Koo, this paper has been made available for downloading in Adobe Acrobat format from the IICS - Minnesota website at:

http://www.IICS-MN.org