University of Alabama in Huntsville
LOUIS

Honors Capstone Projects and Theses

4-22-1996

Software Engineering and Hypermedia Application Design: The Classic Life Cycle vs. Relationship Management Methodology

Stacy Lukins

Follow this and additional works at: https://louis.uah.edu/honors-capstones

Part of the Software Engineering Commons

Recommended Citation
https://louis.uah.edu/honors-capstones/105

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.
Software Engineering and
Hypermedia Application Design:
The Classic Life Cycle vs.
Relationship Management
Methodology

Stacy Lukins
Advisor: Tony Orme
# TABLE OF CONTENTS

**LIST OF FIGURES**

**PROLOGUE**

**INTRODUCTION**

**BACKGROUND / LITERATURE REVIEW**

**APPROACH**

**RESULTS**

- The Classic Life Cycle and WebJUNK
- The Relationship Management Methodology and WAY-FM
- The Classic Life Cycle vs. RMM
  - Project Documentation
  - Prototyping
  - User Interface Focus
  - Hyperlink Representation
  - Script Representation

**CONCLUSION**

**WORKS REFERENCED**

**APPENDIX A: Classic Life Cycle Data Model**

**APPENDIX B: Relationship Management Data Model**

**APPENDIX C: WebJUNK Web Pages**

**APPENDIX D: WAY-FM Web Pages**
LIST OF FIGURES

FIGURE 1: THE CLASSIC LIFE CYCLE .................................................. 5
FIGURE 2: RELATIONSHIP MANAGEMENT METHODOLOGY ...................... 6
FIGURE 3: LEVEL 00 WEB JUNK DATA FLOW DIAGRAM .......................... 13
FIGURE 4: RMDM DIAGRAM FOR WAY-FM WEB SITE ............................ 15
FIGURE 5: EMPLOYEES SLICE DIAGRAM ........................................ 16
FIGURE 6: PRODUCT INDEX PAGE ................................................... 18
FIGURE 7: LEVEL 02 WEB PAGE PRODUCT PACKAGE DATA FLOW DIAGRAM ......... 19
FIGURE 8: E-R DIAGRAM FOR WAY-FM WEB SITE ................................ 20
FIGURE 9: WAY-FM NEWS SLICE DIAGRAM ...................................... 21
FIGURE 10: NEWSLETTER PAGE ...................................................... 22
FIGURE 11: STATION LOCATIONS PAGE ........................................... 23
FIGURE 12: LEVEL 02 FORM PROCESSING DATA FLOW DIAGRAM ............. 25
FIGURE 13: RMM and CLC COMPARISON ......................................... 26

Stacy Lukins
Efficiently developing software products has been an important concern across the years of software development. Many organizations select one methodology and follow it exclusively in order to maintain consistency among all software development projects. However, hypermedia applications differ significantly from traditional software development projects. Therefore, traditional software engineering paradigms may not be well suited to designing hypermedia applications and other methodologies should be examined.
INTRODUCTION

An important part of software engineering is choosing the best paradigm to efficiently develop a software product. A methodology should be chosen based on the nature of the software project, the method and tools to be used, and the demands of the customer. Unfortunately, many organizations promote or require the use of only one software engineering methodology for all software development. While this maintains consistency among software development projects, not all applications can be efficiently developed nor can they be represented well using the same methodology. Inefficient development wastes time and money, and poor representation in documentation makes maintenance costly and difficult. Therefore, for a particular software project, the selection of an appropriate paradigm should be an important part of the design process.

Hypermedia applications, a rapidly increasing segment of the software development market, differ significantly from traditional software development projects. Therefore, traditional software engineering paradigms, such as the Classic Life Cycle (CLC), may not be well suited for this class of applications, and other software development methodologies should be considered. One such alternative methodology, Relationship Management Methodology (RMM), is reviewed in this project.

In the Fall 1995 CS499 course, I participated in the team development of a World Wide Web hypermedia application, WebJUNK, using the Classic Life Cycle paradigm. The WebJUNK project exposed many shortcomings in the Classic Life Cycle paradigm when
used to design a hypermedia project. I feel RMM would have been much more useful and efficient in the development of WebJUNK, and should be seriously considered for the development of other hypermedia applications.

This project provides a comparison of the Relationship Management Methodology and the Classic Life Cycle paradigm to determine whether or not RMM is truly a more appropriate method for designing hypermedia applications than the Classic Life Cycle. At the same time, it stresses the importance of selecting an appropriate software engineering methodology in the development of any software project.
BACKGROUND / LITERATURE REVIEW

The software engineering process consists of three generic phases: definition, development, and maintenance. These three phases are encountered in all software development, regardless of application area, project size, or complexity (Pressman 21). The generic phases are broken down and applied according to software engineering paradigms, each of which approach the application of the phases differently. Choosing a paradigm that is appropriate for the development of a specific software application is an important part of the software engineering process. Using a method well-suited to the project can help save time and money, produce a better quality product, and ensure easier maintenance of the software.

Several methodologies exist for the development of software applications. Among the more traditional is the Classic Life Cycle, the oldest and the most widely used paradigm for software engineering (Pressman 15). The CLC paradigm is outlined by Roger S. Pressman in *Software Engineering: A Beginner's Guide* and is discussed in more detail in Pressman's *Software Engineering: A Practitioner's Approach*. This software engineering paradigm consists of six sequential phases for software development:

+ system engineering
+ requirements analysis
+ design
+ coding
+ testing
+ maintenance

These relationship between these phases are illustrated in figure 1.
The Classic Life Cycle paradigm calls for one stage to be completed in its entirety before the next stage begins; the output of the first stage serves as input to the second stage, whose output serves as input to the third stage, and so on.

![Diagram of the Classic Life Cycle](image)

**FIGURE 1: THE CLASSIC LIFE CYCLE**

The increasing number of hypermedia applications being developed calls for the re-evaluation of traditional approaches to the design of software applications. The first full hypermedia design methodology is discussed by Tomas Isakowitz, Edward A. Stohr, and P. Balasubramanian in "RMM: A Methodology for Structured Hypermedia Design."

RMM consists of the following stages:

- feasibility
- information/navigation requirements analysis
- design
- construction
- testing
- evaluation

Figure 2 illustrates these stages in RMM.
While these stages, at the top level, do not seem much different than those of the Classic Life Cycle, the main difference lies in the methods used for the design and construction phases. RMM focuses on the navigational aspects of hypermedia applications, while the Classic Life Cycle leaves little room for such representation. This project focuses on the E-R Design, Entity Design, and Navigation Design phases of RMM and how they are better suited for hypermedia application development than the corresponding sections of the Classic Life Cycle model.
**APPROACH**

The focus of this honors senior project is the comparison of the RMM and Classic Life Cycle paradigms for the development of hypermedia applications. Specifically, this project is concerned with the design phase of hypermedia application development for the World Wide Web. In order to compare RMM to the Classic Life Cycle, I designed and implemented a hypermedia application for the World Wide Web using RMM. I compared the design and development of the new application to that of the WebJUNK system previously developed using the Classic Life Cycle paradigm. The advantages and disadvantages of each software engineering paradigm were noted during the various development efforts and are detailed in this report.

For the new application, I have

+ analyzed and determined requirements (requirements are outlined only; no formal requirements document will be produced)
+ designed the application
+ constructed the application
+ tested and evaluated the resulting product

Throughout this process, I recorded observations about the differences (both strengths and weaknesses) between RMM and the Classic Life Cycle paradigms in the design and implementation of a hypermedia project. Resulting discussions of these paradigms as applied to hypermedia design include the following topics: appropriateness, ease of use, problems encountered, and the strengths and weaknesses of each paradigm. In my
In the opinion, the comparison also demonstrates the importance of selecting an appropriate method for developing a software application.

Included in the project are:

- documents detailing the RMM design process for the product.
- disk copy of the hypermedia project (Web pages, source code for scripts, etc.)
- a comparison of RMM and the Classic Life Cycle methodologies when used to develop hypermedia projects and any conclusions as to the most appropriate method (perhaps a combination of methods).
RESULTS

I have used both the Classic Life Cycle and the Relationship Management Methodology in the design of hypermedia applications. The following sections detail the strengths and weaknesses of each method applied to hypermedia application design.

The data model defined by the Classic Life Cycle paradigm, as outlined by Pressman, can be found in Appendix A. Appendix B describes the Relationship Management data model presented by Isakowitz, Stohr, and Balasubramanian.

The Classic Life Cycle and WebJUNK

The Classic Life Cycle paradigm was used in the development of the WebJUNK system, a World Wide Web site for the fictitious Johnny Unites New Kind (JUNK) RV company. The JUNK company was interested in advertising their RV products on the internet through a series of Web pages displaying their products’ main features. JUNK also wanted to provide a guestbook form to track customer interest and to allow customers to request additional product information by mail. In addition, JUNK was interested in providing an intranet, including on-line information such as memos and employee manuals, strictly for internal company use.

WebJUNK consists of many Web pages designed to fulfill JUNK’s needs. These Web pages are included in Appendix C. WebJUNK also contains CGI scripts for parsing and processing the customer information entered into the guestbook form.

Stacy Lukins
The Relationship Management Methodology and WAY-FM

The WAY-FM World Wide Web site was designed using the Relationship Management Methodology. WAY-FM is a listener-supported Christian radio station targeted towards today's youth. The purpose of the WAY-FM site was to advertise the WAYM radio station on the internet, providing information on hit lists, employees, programming, supporters, and more. WAY-FM wanted to recognize those who contribute to the station and were interested in providing links to other great Christian sites, bringing together the Christian Web community. WAY-FM was also interested in advertising station locations and project information. The WAY-FM Web site consists of many pages designed to fulfill these Web requirements for WAY-FM. These pages are included in Appendix D.

The Classic Life Cycle vs. RMM

This comparison of the Relationship Management Methodology and the Classic Life Cycle for the development of hypermedia applications includes the following topics:

- Project Documentation
- Prototyping Support
- User Interface Focus
- Hyperlink Representation
- Script Representation

Project Documentation

Project documentation is important for any software product, since extensively documented products are easier to maintain and expand. When people responsible for the maintenance of the software product are studying the code several years after it was first

Stacy Lukins
produced, documentation is helpful to refer to in case some aspect of the design or
implementation is unclear. Documentation also provides a representation of the project at
a high level, so one can observe the relationships among the software modules and
external entities – relationships which are often hard to discover by tracing through a
large amount of code.

Both the Classic Life Cycle and the Relationship Management Methodology provide
good documentation capabilities. The CLC lays out a detailed description of all project
documents that should be produced and the content of those documents. This paradigm
requires the generation of a software specification document, a software design
document, and a test plan document. Similarly, RMM details both the types and the
content of documents that should be produced during the software development process.
RMM calls for a feasibility document, a requirements document, and a design document.

Prototyping

An important part of hypermedia design involves organizing the information to be
presented so that it is clear and accessible to the user. The functionality and appearance
of a Web site is essential to its success, since a user can easily access thousands of other
locations. Typically, a great deal of time is invested in the organization and presentation
of the information in the Web application. Prototyping can save time and money in Web
site development by allowing the clients to see a sample version of the product, in which
the functionality is perhaps represented but not yet fully implemented, so changes can be
made as the software system develops.

Stacy Lukins
The Classic Life Cycle does not allow for prototyping, since the CLC steps must be followed sequentially and one step must be entirely completed before the next step is started. This can be problematic if the client is unhappy with the product results and requests changes, resulting in the loss of a great amount of work.

RMM, on the other hand, does allow for prototyping of the Web project before it is completely constructed; it recognizes that prototyping and user testing is important to ensure that the look and functionality of the Web system will be satisfactory upon completion.

**User Interface Focus**

The success of a hypermedia application greatly depends on its appeal to the user. The information in the Web site should be formatted and organized in an easy to understand manner, so the user does not get frustrated with the site and go somewhere else, never to return again. Therefore, the presentation of the information on the Web site is a very important concern.

The Classic Life Cycle design phase focuses on system interfaces rather than user interfaces. In other words, CLC is concerned with how the information is passed between the software and system components, rather than how the information is organized and presented to the user. In the CLC data model, the user is not typically represented, because he or she is too far removed from the system. Instead, the model focuses on the interaction between the software package and various system components, such as a

*Stacy Lukins*
control panel, monitor, or printer. These system components either send data to or receive data from the software package. The transformations of the data within the software package and the interaction with system components is represented by CLC data flow diagrams; transformations which are not observed by the user.

In the WebJUNK system, for example, the software package interfaces with the Netscape server on which the software is stored. This is represented in figure 3, the top level data flow diagram for WebJUNK. The system components include the Netscape Server and the Email Interface (for sending email responses to the user). Passed between the software and these components are URL requests, guestbook form data, formatted Web pages, and email responses. The Netscape server performs any interaction needed with the user’s computer system (sending the Web pages to the user’s Web browser, for example), and this interaction is not represented in the diagrams.

**FIGURE 3: LEVEL 00 WEB JUNK DATA FLOW DIAGRAM**
RMM, on the other hand, focuses on the user interface. The Relationship Management Data Model (RMDM) is concerned not only with what data the user sees, but how the data is presented to the user. The RMDM diagram, the final product of the three stages in the design process, represents how the user will navigate the hypermedia application. Figure 4 shows the RMDM diagram for the WAY-FM system (OK, it looks a little scary at first).

From the home page, the user can access WAY-FM News, Employees, the Top Ten List, WAY-FM Supporters, Programming, and the Calendar of Events. However, Concert Connections, Church Events, and Employee Picks can only be accessed through other entities (e.g., Concert Connections can only be accessed through Calendar of Events). The diagram also shows the way in which the user will enter each entity; for example, the user enters the Calendar of Events entity through an indexed guided tour. In addition, the user can move between certain entities. For example, the user can enter the Programming entity from the Employees entity via an indexed guided tour with the condition hosts, so that only those programs hosted by the current employee are displayed.
The RMDM directly represents what the user will see and at what point in the system they will see it. RMM goes down to an even lower level than the RMDM diagram, describing with slice diagrams exactly what information will be presented inside each entity. The slice diagram for Employees is displayed in figure 5. The Employee entity contains general information, a bio sketch, family information, and contact information. In addition, each slice is broken down and the attributes included in each slice are specified (for example, the contact info slice contains the employee's name, phone number, fax number, and email address).
Overall, RMM is not concerned with how the information will get to the user (since that is taken care of by the Netscape server), but with what information the user will see, how the information is organized, and how the user can get additional information.

Hyperlink Representation

Hyperlinks are a main component, perhaps the most important component, of hypermedia applications, and they should be represented by the data model used to design any hypermedia application. This is the principle difference between the Classic Life Cycle and the Relationship Management Methodology data models.

The Classic Life Cycle fails to provide for the representation of hyperlinks between the Web pages in the software package. In the CLC data model, the Web pages in a hypermedia application never actually interact with one another, thus links between pages
are not needed in the representation. When a hyperlink is accessed by the user, their
computer system issues a URL request to the Web server where the pages are stored; the
Web server retrieves the requested page and sends the page back to the user’s system.
The Web server does not care which page contained the hyperlink to the requested page;
it only knows the user has requested another page. Therefore, only the page’s interaction
with the Web server is represented in Classic Life Cycle data flow diagrams.

In the WebJUNK system, the Product Index Page, illustrated in figure 6, provides the
user with links to the guestbook page and all product pages (organized by price and size).
However, in the Level 02 Product Page Package data flow diagram displayed in figure 7,
these links between the pages are not represented in any way. The data flow diagram
shows no relationship between any of the pages – just the interactions between the pages
and the Netscape server. The URL Request comes from the Netscape server and the
Formatted page information is sent to the Netscape server.
technical specifications together with available options and an itemized price list by e-mail or conventional mail if you prefer. On the guestbook page you may also request to receive a complete list of the RVs we have available in a particular type, size, or price category. After you have reviewed the list, return to this page and sign into the guestbook page to obtain more detailed information on any one individual product.

Mark this page with a bookmark now so you can easily find it again!

If you have any questions which cannot be answered on these web pages, you may call us at (800) 123-4567 between the hours of 8.00 a.m. and 6.00 p.m. EST.

---

### Products according to price range

- **$10,000 to $20,000**
  - Coachmen Model 200 RB, Trailer, small
  - Coachmen Model 210 MB, Trailer, medium
  - Coachmen Model 248 TB, Trailer, large
  - Coachmen Model 200 RK, Fifth Wheel, small

- **$20,000 to $45,000**
  - Coachmen Model 364 TBS, Fifth Wheel, large
  - Dutchmen Model LTD 33 RL-LE, Fifth Wheel, small
  - Triple E Model 3101, Fifth Wheel, medium

- **$45,000 to $75,000**
  - King of the Road Model 36 CL, Fifth Wheel, large
  - Triple E Embassy Model A31, Motor Home, medium
  - Triple E Regency Model 351, Motor Home, medium

### Products according to size

- **Small**
  - Coachmen Model 200 RB, Trailer, low
  - Coachmen Model 200 RK, Fifth Wheel, low
  - Dutchmen Model LTD 33 RL-LE, Fifth Wheel, medium

---

**FIGURE 6: PRODUCT INDEX PAGE**

Stacy Lukins 18
In contrast to the CLC model, RMM provides an excellent representation of links between pages. In the WAY-FM Web site, the following entities exist:

+ WAY-FM News
+ Employees
+ Employee Picks
+ Top Ten List
+ Calendar of Events
+ Concert Calendar
+ Church Events
+ Supporters
+ Programming

The relationships between these entities are represented in the WAY-FM E-R Diagram in figure 8. For example, the programming and supporters entities are connected through the relationship supported_by and supports; a program is supported_by a certain supporter, and a supporter supports a certain program.
Slice diagrams represent the information presented within an entity. For example, within the WAY-FM news entity (shown by the slice diagram in figure 9) are the slices newsletter, WAY-FM history, station locations, talk to us, prayer request, and project updates. The newsletter slice is the head of the entity, or the entry point. Therefore, when the user selects a hyperlink to WAY-FM news, the first page displayed is the newsletter page, shown in figure 10. From this page, the user can access any of the other slices (as shown in the newsletter page by hyperlinks and in the slice diagram by lines drawn between the slices).
Because the lines in the slice diagram are bi-directional, the user can return to the newsletter page from any of the slices. For example, from the Station Location Web page displayed in figure 11 the user can select the Back to Newsletter link to return to the newsletter page.

Next, an E-R+ diagram should be produced where the entities in the E-R diagram are replaced by their slice diagrams. However, because the E-R+ diagram tends to look cluttered, it is not included in this paper. Finally, everything is put together in the RMDM diagram we saw earlier in figure 4, where often the relationships from the E-R diagram become hyperlinks between entities.

One problem with the RMDM is that hyperlinks to outside sites are not represented in any of the design diagrams. Overall, however, RMM provides an excellent representation of hyperlinks between pages in the Web site.
WAY ~ FM News!

What's happening at your FAVORITE radio station!

- Newsletter — Monthly newsletter from WAY ~ FM
- Station Locations — Where can you get our signal? Nashville, TN is our home, but we have repeaters all over the Mid-South! We even have a satellite link!
- History — How did WAY ~ FM get started?
- Prayer Requests — Requests from the Station and Listeners. Send us yours!
- Project Updates — Find out about the Power Increase and more!

RETURN TO WAY-FM HOME.

FIGURE 10: NEWSLETTER PAGE

Stacy Lukins
WAY ~ FM Station Locations

WAY ~ FM studios are located in Brentwood, Tennessee, a suburb of Nashville. We can be heard in Nashville and in cities throughout the mid-South, thanks to repeaters and a digital-quality satellite uplink! Check out our stations in Alabama, Georgia, and Tennessee!

Our FM radio signal can be found in the following locations on these frequencies:

**Alabama**
- Florence, 99.5
- Huntsville, 88.7
- Killen, 89.7

**Georgia**
- Rome, 90.3

**Tennessee**
- Chattanooga, 101.5 & 104.7
- Clarksville, 88.1
- Cleveland, 91.3
- Cookeville, 90.1
- Donelson, 99.3
- Fayetteville, 95.7
- Lawrenceburg, 105.3
- Nashville, 88.7 (increasing our power from 3,000 to more than 16,000 Watts very soon!)
- Pulaski, 106.7

**New Areas!**

WAY ~ FM has received permits from the FCC to expand our coverage to Kentucky and Dalton, Georgia. Please pray for us as we try to find sponsors in those areas to pay for equipment installation.

We are constantly trying to expand our coverage area. If you would like to hear WAY ~ FM radio in your area, please contact our General Manager, Bob Augsburg.

Back to News

RETURN TO WAY-FM HOME

FIGURE 11: STATION LOCATIONS PAGE
Script Representation

Scripts (such as CGI and Java) are an important part of hypermedia applications. Scripts provide the application with much of its enhanced functionality, such as searching capabilities and the ability to dynamically generate Web page content. Therefore, to fully represent the hypermedia application design, scripts should be included in design elements.

Since scripts are very similar to traditional software programs, they fit nicely into the Classic Life Cycle data model. Data flow diagrams in the data model represent the steps the data must go through in order to be transformed into the proper format to be sent to a system component or stored in a database. With the data flow diagrams, one can see exactly where data is modified, what data is modified, and which modules are responsible for certain tasks.

In WebJUNK, for example, one can look at the Level 02 Form Processing data flow diagram, shown in figure 12, to quickly and easily determine which module outputs the form response message to the system (Generate Success/Fail Form Received Message). This is much easier than tracing through many lines of code, especially if the programmer used cryptic module names. In addition, one can use the data flow diagram to trace the path of any data piece from start to end.
The Relationship Management Methodology, while providing excellent representation of hyperlinks in the World Wide Web system, provides no method of representing scripts. Though scripting is perhaps not the most important aspect of hypermedia design, it is still important and should be taken into account in project documentation. Failure to represent the scripts leaves the hypermedia application design incomplete, and RMM seemingly ignores this aspect.
CONCLUSION

Figure 13 summarizes the strengths and weaknesses of the Relationship Management Methodology and the Classic Life Cycle paradigm as they apply to hypermedia application design, and especially to Web site design.

<table>
<thead>
<tr>
<th></th>
<th>RMM</th>
<th>Classic Life Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Documentation</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>User Interface Focus</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Hyperlink Representation</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Prototyping Support</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Script Representation</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

**FIGURE 13: RMM and CLC COMPARISON**

If the use of only one method is permitted in the design of hypermedia projects, RMM is the better method; however, RMM has a serious shortcoming in the representation of scripts. Ultimately, a combination of the two methods may be the best alternative. In that way, the strengths of both methods can be combined to give a strong representation of a hypermedia application. For Web site design, I recommend the use of RMM augmented by script representation methods defined by the Classic Life Cycle. In my opinion, this combination provides the most complete representation of the hypermedia application.
WORKS REFERENCED


Lukins, Stacy, Leigh Kelley, Astrid McIntosh, and Michael McEniry. WebJUNK Documentation.
APPENDIX A: Classic Life Cycle Data Model

from Software Engineering: A Beginner's Guide

Stacy Lukins
Develop a flow model for the software.

Information is transformed as it flows through a computer-based system. The system accepts input in a variety of forms and applies hardware, software, data base, and human elements to transform input and produce output in a variety of forms. Input may be a control signal transmitted by a transducer, a series of numbers typed by a human operator, a packet of information transmitted on a network link, or a voluminous data file retrieved from disk storage. The transform(s) may comprise a single logical comparison, a complex numerical algorithm, or the rule-inference approach of an expert system. Output may light a single LED or produce a 200-page report. In effect, a data flow model can be applied to any computer-based system, regardless of size and complexity.

One method for representing information flow through a computer-based system is illustrated in Figure 3.7. The overall function of the system is represented as a single information transform, noted as a bubble in the figure. One or more inputs, shown as labeled arrows, drive the transform to produce output information. It should be noted that the model may be applied to the entire system or to the software element only. The key is to represent the information fed into and produced by the transform.

Use data flow diagrams to indicate how data move through the software.

As information (objects that can be categorized as data and control items) moves through software, it is modified by a series of transformations. These transformations are analogous to the processing operations discussed earlier in this chapter. A data flow diagram (DFD) (also called a data flow graph or bubble chart) is a graphical technique that depicts data flow and the transforms that are applied as data move.

* It is important to note that there are many different methods that may be used to represent software requirements. The method presented in this book is derived from an approach called Structured Analysis. Other methods, such as Data Structured Systems Development (DSSD), System Analysis and Design Technique (SADT), and Jackson System Development (JSD) are used by software engineers. For additional references, see the Further Readings section at the end of this chapter.
Figure 3.7 Representing Information Flow

from input to output. The basic form of a data flow diagram is illustrated in Figure 3.8.

Data or control items (data1) are produced by an object (originator1) that is an information source. Data1 flows into the system and is operated upon by transform1, which produces another data item, data2. Additional data objects are created through the application of other transforms or by retrieval from a data store. Ultimately, output, in the form data6, is sent to an information sink, receiver1.

The data flow diagram (DFD) may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. A level 01 DFD, also called a fundamental system model, represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Additional transforms and information flow paths are represented as the level 01 DFD is partitioned to reveal more detail.

DFD notation is illustrated in Figure 3.9. A rectangle is used to represent an external object, that is, a system element (e.g., hardware, a person) or another system that produces information for transformation by the software or receives information produced by the software. A circle or oval represents a process or transform that is applied to data and changes it in some way. An arrow represents one or more
Figure 3.8 Data Flow Diagram

data items. All arrows on a data flow diagram should be labeled. The double line represents a data store — stored information that is used by the software. DFD symbology is exceptionally simple and is one reason that data flow-oriented analysis techniques are the most widely used.

---

**Figure 3.9 Data Flow Diagram Notation**
ANALYZING A PROBLEM

It is interesting to note that the objects determined as part of earlier steps correspond to boxes (information sources and sinks), arrows (data items), and double lines (data stores) on a DFD. Operations correspond to circles (transforms). Therefore, the object and operation tables provide the basic information required to derive data flow diagrams.

As a simple example of a data flow diagram, consider information flow for a typical telephone call (Figure 3.10). The level 01 DFD for a telephone call indicates that output is the sound of the caller's voice received by the listener. Input to the telephone call is the caller's voice and a keyed phone number. Figure 3.10b illustrates a level 02 refinement of the level 01 DFD. As shown in the figure, more information about both data flow and process function (transforms) is provided. The caller's action of depressing the key pad is transformed by associated electronics into a series of audible frequencies (tones). The frequencies flow to a switching system that performs requisite routing and establishes a link from sender to receiver. The sound of a human voice is transformed by a vibration transducer that produces a signal.
as output. The switching system moves the voice signal to a receiver that transforms the signal back to sound. It is important to note that the data flow diagrams for this example represent an entire system, not just software. In general, we use DFDs to represent only the software element of a system.

Although the above example is a gross oversimplification, the flow of information, represented by the data flow diagram, is easy to discern. Each transform in the diagram could be refined still further to provide greater detail about key pad processing, transducers, or the switching system. That is, the diagram may be layered to show any desired level of detail.

It is important to note that no explicit indication of the sequence of events (e.g., is phone number keyed before or after voice input?) is supplied by the diagram. Procedure or sequence may be implicit in the diagram, but explicit procedural representation is generally delayed until software design.

As we noted earlier, each of the bubbles may be refined or layered to depict more detail. Figure 3.11 illustrates this concept. A fundamental model for system F indicates that primary input is A and final output is B. We refine the F model into transforms f1 to f7. Note that information flow continuity must be maintained, that is, input and output to each level of refinement must remain the same. Further refinement of f4 depicts detail in the form of transforms f41 to f45. Again, the input (X,Y) and output (Z) remain unchanged.

The data flow diagram is a graphical tool that can be very valuable during software requirements analysis, establishing a foundation for subsequent software design. However, the diagram can cause confusion if a software engineer attempts to design a program using data flow notation. Software design requires a specification of program architecture, followed by a representation of the procedural details of program components (see Chapter 4 for an extended discussion of design). Because the data flow diagram notation cannot be applied to the representation of program architecture or processing details (e.g., loops, conditions), a DFD is inadequate as a design tool.

A few simple guidelines can aid immeasurably during derivation of data flow diagrams: (1) the level 01 data flow diagram should depict the software/system as a single bubble; (2) primary input/output/files should be carefully noted; (3) all arrows and bubbles should be labeled with meaningful names; (4) information flow continuity must
APPENDIX B: Relationship Management

Data Model

from "RMM: A Methodology for Structured Hypermedia Design."

Stacy Lukins
Table 1 illustrates the usefulness of the RMM approach for design and development of hypermedia applications. The two axes representing the structure and volatility of the information are really continua rather than discrete branches of dichotomy. Applications in the two domains we mentioned—product catalogs and hypermedia front-ends of databases or legacy applications—are highly structured and have high information volatility, making the RMM methodology particularly appropriate. At the opposite end of the spectrum, an artistic work may not have a readily discernible structure and usually remains unchanged over time. In this case, RMM is not applicable. Highly structured applications that remain unchanged over a long period of time can benefit from the RMM methodology during the design and construction phases but do not require much maintenance, so that the updating problem is relatively unimportant and the advantage to be gained by the RMM approach is not as pronounced.

Finally, applications that have irregular (or dynamic) structures and high volatility may gain little from the use of the RMM approach. In this case, however, we reserve judgment as it is possible that some parts of the domain may be structured and that the problem of high information volatility can be at least partially addressed.

A number of commercial products, including Documentum by Documentum Inc., PDM by Xyvision Inc., RDM by Interleaf, and SGML Server by Information Dimensions, use an approach similar to RMM in that they provide flexible access mechanisms to documents that are stored in a database. The approaches used in these products are proprietary and have not been reported in the research literature. Moreover, these systems do not provide support for the design process, which is the focus of RMM.

**Relationship Management Data Model (RMDM)**

A data model is a set of logical objects used to provide an abstraction of a portion of the real world. Data models are necessary to express an application's design. A number of researchers have developed data models for hypertext systems. For example, [18, 24] Dexter [11, 12], HB1 and HB3 [16, 20], Trellis [23]. However, it is important to differentiate a hypertext system from a hypertext application. The former is an environment that facilitates the creation of the latter. A data model for a hypertext system [1, 7, 20, 24] details its internal architecture but is of little value in modeling hypermedia applications. This is because describing the layout of a general purpose engine is quite different from modeling an application domain; a different kind of data model is needed for this purpose. In our case, RMDM provides a language for describing the information objects and the navigation mechanisms in hypermedia applications.

Database models are useful abstractions in database applications, but the peculiarities of hypermedia, in particular the navigation aspect, require new models. Garzotto, Paolini, and Schwabe's HDM data model ([9]; see also Garzotto, Paolini, and Mainetti in this issue) is appropriate for describing the structure of the application domain, and we base our data model on HDM and its successor, HDM2 [8]. HDM and HDM2 describe representation schemes but provide little information on the procedures for using

![](image.png)

**Figure 1.** The Relationship Management Data Model (RMDM) primitives. E-R primitives model how information is structured in the application domain. The slice domain primitive models how information is to be presented. The access primitives model navigation.
Table 1. Usefulness of RMM approach

<table>
<thead>
<tr>
<th>Volatility of Information</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Medium usefulness</td>
<td>High usefulness</td>
</tr>
<tr>
<td></td>
<td>[e.g., Kiosk Application]</td>
<td>[e.g., Product catalog, DBMS interface]</td>
</tr>
</tbody>
</table>

Table 1. Usefulness of RMM approach

those representations in the design process; that is, they do not describe a hypermedia design and development methodology. Lange [15] and Schwabe and Rossi (in this issue and [22]) study object-oriented approaches to hypermedia design. Recent works by us [2] and by Garzotto, Mainetti, and Paolini [10] focus on this problem. The present article represents an update and expansion of [2]. Our methodology differs from that presented in [10] in several dimensions, including the recommended sequence of steps, additional access structure formalisms, increased emphasis on graphic representations, and a more detailed, step-by-step procedure for hypermedia design and development. The proceedings of two recent hypermedia design workshops [14, 21] discuss various other issues arising in hypermedia design.

Here, we describe the RMDM data model which is the cornerstone of the RMM methodology. Figure 1 shows RMDM's modeling primitives. In the upper part of the figure are the domain primitives, which model information about the application domain. Entity types and their attributes represent abstract or physical objects, such as person or bank account. Associative relationships, which can be one-one or one-many, describe associations among different entity types. As in database modeling, a many-many relationship is factored into two one-many relationships.

Because entities may consist of a large number of attributes of different natures (e.g., salary information, biographical data, photographs), it may be impractical or undesirable to present all of the attributes of an entity instance at once. Thus, attributes are grouped into slices. For example, a person entity with attributes name, age, picture, and biography may have a general slice, containing name, age, and photograph, and a biography slice, with name and biography. Hence, each instance of the entity "person" will be presented by two slices, and, if the application supports it, a user may choose which one to view. The graphical notation for slices—meant to resemble a pizza slice—appears in the middle of Figure 1.

Navigation is supported in RMDM by the six access primitives shown at the bottom of Figure 1. The unidirectional and bidirectional links are used to specify access between slices of an entity. It is important to stress that these links can be used to navigate only within the boundaries of an entity. RMDM supports navigation across different entities via indices, guided tours, and groupings. An index acts as a table of contents to a list of entity instances, providing direct access to each listed item. A guided tour implements a linear path through a collection of items, allowing the user to move either forward or backward on the path. There are a number of useful variations on guided tours. For

Figure 2. Examples of the conditional RMDM access constructs.
example, a circular guided tour links the last element back to the first: a guided tour with return to main has a distinguished node that contains information about the guided tour itself (e.g., “This is a guided tour of faculty homepages”) and is both the starting and the ending point of the tour; and a guided tour with entrance and exit has different entrance and exit nodes. RMM is capable of accommodating all of these variations on guided tours. However, for the purposes of this article, it suffices to consider a generic guided tour construct, shown in the middle of Figure 1.

The grouping construct is a menu-like mechanism that enables access to other parts of a hypermedia document. A typical example of a grouping is the opening screen of many applications, which serves the purpose of providing access to other features, such as indices and guided tours. Indices are special kinds of groupings. We are currently investigating other useful grouping constructs, such as the multilevel hierarchical structures so common in knowledge classification schemes.

The conditions or logic predicates qualifying indices and guided tours determine which instances of an entity are accessible from the construct. For example, Figure 2a shows a conditional guided tour of all associate professors. The predicate faculty (rank = 'associate') indicates that only those entity instances of faculty whose rank attribute is associate participate in the guided tour. The right part of the figure shows an instance of such a guided tour. Figure 2b is an example of a conditional index web. Here, access is granted via an index-like construct. There are also return links from each participating node to the index, as shown on the right of the figure. Last, conditional indexed guided tour webs combine indices and guided tours to provide a richer access structure.

We will use the Stern School’s Information Systems Department handbook application (ISweb) as a basis for discussion (URL: http://is-2.stern.nyu.edu/isweb). V. Balasubramanian et al. present a different application, also developed with RMM, in this issue. The ISweb handbook contains descriptions of the graduate programs and courses offered by the information systems (IS) department and a list of faculty members in the department and their research interests. We have chosen this application for various reasons: (1) Many readers are
acquainted with this domain (2) Its moderate complexity enables an illustration rich in details (3) It illustrates the problems that arise when updates are relatively frequent (semiannual in this case).

Figure 3 shows the complete RMDM diagram for the ISweb application. Note that, in contrast to an entity-relationship diagram that represents the design of a database, an RMDM diagram describes how users will navigate a hypermedia application. To avoid cluttering, slices are not included in Figure 3, and only the key attributes of entities are shown. At the top of the figure the grouping mechanism implements a main menu. Access into the faculty and course information is provided via guided tours; access into programs by means of an index. On choosing the guided tour to the faculty entity, the user can move back and forth among all faculty members (ordered alphabetically). From the faculty entity, there is a conditional index into courses with predicate teaches(F, C). The reciprocal index taught by(C, F) can be accessed from courses. Together, these two indices represent a many-many relationship between faculty and courses.

The teaches conditional index allows the user to move from the faculty entity to the courses taught by that faculty member. Had we provided an indexed guided tour for the courses taught by a faculty member, the user would be able to choose which course to visit first and from there, he or she could use the next and previous links to visit other courses taught by the same faculty member.

The Relationship Management Methodology (RMM)

The RMM methodology is shown graphically in Figure 4 within the context of the complete software development cycle. RMM focuses on the design, development, and construction phases. In this article we concentrate on the design of access mechanisms, which is achieved through the first three steps of the methodology (shown in the shaded area of Figure 4). Although feasibility, requirements analysis, and testing are undeniably important phases in software development, they are beyond the scope of this article. To evaluate hypermedia applications, one can use techniques like those proposed by Garzotto, Paolini, and Mainetti (in this issue) and by Botafogo, Rivlin, and Schneiderman [3].

The labels on the arrows in Figure 4 represent the various intermediate artifacts generated through the use of the methodology. Although present in the methodology, we do not show the feedback loops among the remaining stages to avoid cluttering the figure. Feedback loops between the RMM design stages are shown by dashed lines.

The RMDM data model provides a strong prescription for choosing the nodes and links in the hypermedia application. However, many design issues must be decided independently by the designer (see Thüring, Hannemann, and Haake in this issue). While our main purpose here is to outline the design methodology, we also discuss some design guidelines for each step.

**Step S1: E-R Design.** The first design step is to represent the information domain of the application via an
APPENDIX C: WebJUNK Web Pages

Stacy Lukins
Product Index Page

To proceed directly to the listed features click on these items:

- How to use our product index page
- What you will find on our product pages
- How to get detailed product information
- Products according to price
- Products according to size
- Guestbook page

How to use our product index page

Our product index page allows you to view only those products which are of interest to you. By selecting one of the buttons below, you may choose to look at the RVs in a particular price or size category. Once on the product page, you can view the next RV in the category you selected by clicking on "Next in price" or "Next in size" at the bottom of the page.

Next to the buttons you will find a brief listing of all RVs in the category together with their types (taller, fifth wheel, or motor home). You may choose to look at any product page directly by clicking on the hyperlink in the listing. This will enable you to select only a certain type of vehicle and the button label will give you information about its size and price range. For easy cross-reference, the size of the RV is listed in the price category, and the price range appears in the listings for the size category.

What you will find on our product pages

Each of our product pages contains an exterior picture of the RV and a floorplan of the advertised model. Some pages also show an interior view of the vehicle. Selected standard and optional features of the RV will be listed and the price range will be narrowed down within the price category.

How to get detailed product information

Our product pages provide only a sample of the RVs we have available. If you would like more detailed information about any of these vehicles, sign our guestbook page and we will send you the technical specifications together with available options and an itemized price list by e-mail or conventional mail if you prefer.
On the guestbook page you may also request to receive a complete list of the RVs we have available in a particular type, size, or price category. After you have reviewed the list, return to this page and sign into the guestbook page to obtain more detailed information on any one individual product.

Mark this page with a bookmark now so you can easily find it again!

If you have any questions which cannot be answered on these web pages, you may call us at (800) 123-4567 between the hours of 8:00 a.m. and 6:00 p.m. EST.

Products according to price range

- Coachmen Model 200 RB, Trailer, small
- Coachmen Model 210 MB, Trailer, medium
- Coachmen Model 248 TB, Trailer, large
- Coachmen Model 200 RK, Fifth Wheel, small

- Coachmen Model 364 TBS, Fifth Wheel, large
- Dutchmen Model LTD 33 RL-LE, Fifth Wheel, small
- Triple E Model 3101, Fifth Wheel, medium

- King of the Road Model 36 CL, Fifth Wheel, large
- Triple E Embassy Model A31, Motor Home, medium
- Triple E Regency Model ?, Motor Home, medium
Products according to size

- **small**  
  sleeps 4
  - Coachmen Model 200 RB, Trailer, low
  - Coachmen Model 200 RK, Fifth Wheel, low
  - Dutchmen Model LTD 33 RL-LE, Fifth Wheel, medium
  - Triple E Regency Model 9, Motor Home, high

- **medium**  
  sleeps 6
  - Coachmen Model 210 MB, Trailer, low
  - Triple E Model 3101, Fifth Wheel, medium
  - Triple E Embassy Model A31, Motor Home, high

- **large**  
  sleeps 8
  - Coachmen Model 248 TB, Trailer, low
  - Coachmen Model 364 TBS, Fifth Wheel, medium
  - King of the Road Model 36 CL, Fifth Wheel, high

---

**WebJUNK**
Johnny Unites New Kind
Last modified: 1995 Nov 17 14:59:54
King of the Road Model 36 CL

Size: large
Price Range: high
Vehicle Type: fifth wheel

To proceed directly to the listed features click on these items:

- Standard features
- Optional features
- Interior picture
- Floor plan

This King of the Road fifth wheel has an overall length of 375" and sleeps up to 8. Double slide-out rooms add plenty of extra floorspace.

Standard features include
- Deluxe cabinetry with oak raised panel and etched glass doors
- Swivel rocking chair in living room
- AM/FM stereo with cassette and four speakers
- Entertainment center cabinet
- Standup bedroom
- One piece fiberglass tub/shower combination with skylight over tub
- Large double door refrigerator
- Deluxe microwave oven

Some optional features are
- 42" ceiling fan in living area
- Skylight in living room or kitchen
Floor plan

To obtain a complete list of standard and optional features and technical specifications, please sign our guestbook page.
WELCOME TO

Johnny Unites, Inc.

COMPANY RESOURCES

- WWW Site
  Home Page- Product Index

- Site Statistics
  Usage- WWWStat

- Customer Feedback

- Company Manuals
  Employee Handbook- Site Manual

- The RV Industry
  RV Highway- Used RVs- Repo Net RVs- Holiday Rambler- Coachmen

SLAM! Software Solutions
Email: SlamSupport@www.slam.com
This area lists core product ramp pages. Currently that includes only the home page and the product index page.

- [Junk Server Home Page]
- [Product Index Page]

This area simply lists all of the different Recreational Vehicles that we presently have on-line in the WebJUNK package. The following categories of RV's are represented:

- [Model 200 RB; Small, Trailer]
- [Model 210 MB; Medium, Trailer]
- [Model 248 TB; Large, Trailer]
- [Model 200 RK; Small, Fifth Wheel]
- [Model 364 TBS; Large, Fifth]
- [Model LTD 33 RL-LE; Small, Fifth Wheel]
- [Model 3101; Medium, Fifth Wheel]
- [Model 36 CL; Large, Fifth Wheel]
- [Model Emassey; Medium, Motor Home]
- [Model Regency; Medium, Motor Home]

SLAM! Software Solutions
Email: SlamSupport@www.slam.com
This program processes a sequence of NCSA httpd 1.2 or later access_log files and outputs a log summary in HTML format suitable for viewing on a browser. The program wwwstat does not make any changes to or write any files in the server directories. The following categories are represented:

- Daily Transmission Statistics
- Hourly Transmission Statistics
- Total Transfers by Client Domain
- Total Transfers by Reversed Subdomain
- Total Transfers from each Archive Section

This program also maintains usage statistics for the WebJUNK server. We included it also, because we like the top ten site listing! It updates the following information, week by week:

- Total server usage, each week
- The top ten sites by frequency of access, each week
- The top ten documents accessed
- A big graph of server usage over many weeks
- An icon version of the graph available for use in Web pages
Server Access Breakdown for WebJUNK

Last updated: Sun, 05 Nov 1995 16:26:42 (GMT -0600)

- Daily Transmission Statistics
- Hourly Transmission Statistics
- Total Transfers by Client Domain
- Total Transfers by Reversed Subdomain
- Total Transfers from each Archive Section
- Previous Full Summary Period

Totals for Summary Period: Oct 10 1995 to Nov 5 1995

| Files Transmitted During Summary Period | 1026 |
| Bytes Transmitted During Summary Period | 6817333 |
| Average Files Transmitted Daily | 73 |
| Average Bytes Transmitted Daily | 486952 |

Daily Transmission Statistics

<table>
<thead>
<tr>
<th>%Reqs</th>
<th>%Byte</th>
<th>Bytes Sent</th>
<th>Requests</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.85</td>
<td>0.18</td>
<td>12080</td>
<td>19</td>
<td>Oct 10 1995</td>
</tr>
<tr>
<td>0.58</td>
<td>0.13</td>
<td>8639</td>
<td>6</td>
<td>Oct 11 1995</td>
</tr>
<tr>
<td>0.88</td>
<td>0.09</td>
<td>6188</td>
<td>9</td>
<td>Oct 13 1995</td>
</tr>
<tr>
<td>0.39</td>
<td>0.04</td>
<td>2408</td>
<td>4</td>
<td>Oct 17 1995</td>
</tr>
<tr>
<td>2.34</td>
<td>1.15</td>
<td>78197</td>
<td>24</td>
<td>Oct 19 1995</td>
</tr>
<tr>
<td>2.14</td>
<td>0.34</td>
<td>23447</td>
<td>22</td>
<td>Oct 26 1995</td>
</tr>
<tr>
<td>7.12</td>
<td>2.22</td>
<td>151427</td>
<td>73</td>
<td>Oct 27 1995</td>
</tr>
<tr>
<td>27.00</td>
<td>9.91</td>
<td>675513</td>
<td>277</td>
<td>Oct 29 1995</td>
</tr>
<tr>
<td>3.22</td>
<td>0.73</td>
<td>49910</td>
<td>33</td>
<td>Oct 30 1995</td>
</tr>
<tr>
<td>4.78</td>
<td>0.84</td>
<td>57546</td>
<td>49</td>
<td>Oct 31 1995</td>
</tr>
<tr>
<td>1.27</td>
<td>0.35</td>
<td>24004</td>
<td>13</td>
<td>Nov 1 1995</td>
</tr>
<tr>
<td>41.62</td>
<td>77.40</td>
<td>5276943</td>
<td>427</td>
<td>Nov 2 1995</td>
</tr>
<tr>
<td>4.68</td>
<td>6.25</td>
<td>426131</td>
<td>48</td>
<td>Nov 3 1995</td>
</tr>
<tr>
<td>2.14</td>
<td>0.37</td>
<td>24900</td>
<td>22</td>
<td>Nov 5 1995</td>
</tr>
</tbody>
</table>

Hourly Transmission Statistics

<table>
<thead>
<tr>
<th>%Reqs</th>
<th>%Byte</th>
<th>Bytes Sent</th>
<th>Requests</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.29</td>
<td>0.10</td>
<td>6925</td>
<td>3</td>
<td>00</td>
</tr>
<tr>
<td>0.19</td>
<td>0.04</td>
<td>2920</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>0.19</td>
<td>0.04</td>
<td>2920</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3.61</td>
<td>2.17</td>
<td>148014</td>
<td>37</td>
<td>12</td>
</tr>
<tr>
<td>16.08</td>
<td>12.32</td>
<td>839734</td>
<td>165</td>
<td>13</td>
</tr>
<tr>
<td>13.65</td>
<td>25.97</td>
<td>1770212</td>
<td>140</td>
<td>14</td>
</tr>
<tr>
<td>9.84</td>
<td>23.33</td>
<td>1590637</td>
<td>101</td>
<td>15</td>
</tr>
</tbody>
</table>
### Total Transfers by Client Domain

<table>
<thead>
<tr>
<th>%Reqs</th>
<th>%Byte</th>
<th>Bytes Sent</th>
<th>Requests</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.36</td>
<td>0.41</td>
<td>27985</td>
<td>14</td>
<td>com US Commercial</td>
</tr>
<tr>
<td>79.92</td>
<td>94.49</td>
<td>6441643</td>
<td>820</td>
<td>aspire</td>
</tr>
<tr>
<td>18.71</td>
<td>5.10</td>
<td>347705</td>
<td>192</td>
<td>unresolved</td>
</tr>
</tbody>
</table>

### Total Transfers by Reversed Subdomain

<table>
<thead>
<tr>
<th>%Reqs</th>
<th>%Byte</th>
<th>Bytes Sent</th>
<th>Requests</th>
<th>Reversed Subdomain</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.71</td>
<td>5.10</td>
<td>347705</td>
<td>192</td>
<td>Unresolved</td>
</tr>
<tr>
<td>1.95</td>
<td>0.30</td>
<td>20350</td>
<td>20</td>
<td>aspire.laser05</td>
</tr>
<tr>
<td>1.66</td>
<td>0.90</td>
<td>61465</td>
<td>17</td>
<td>aspire.laser07</td>
</tr>
<tr>
<td>2.83</td>
<td>0.79</td>
<td>54012</td>
<td>29</td>
<td>aspire.laser10</td>
</tr>
<tr>
<td>0.88</td>
<td>0.09</td>
<td>6188</td>
<td>9</td>
<td>aspire.laser11</td>
</tr>
<tr>
<td>1.95</td>
<td>0.31</td>
<td>21075</td>
<td>20</td>
<td>aspire.laser12</td>
</tr>
<tr>
<td>35.48</td>
<td>17.15</td>
<td>1168899</td>
<td>364</td>
<td>aspire.laser17</td>
</tr>
<tr>
<td>3.80</td>
<td>0.63</td>
<td>43091</td>
<td>39</td>
<td>aspire.laser18</td>
</tr>
<tr>
<td>31.38</td>
<td>74.32</td>
<td>5066563</td>
<td>322</td>
<td>aspire.laser19</td>
</tr>
<tr>
<td>1.27</td>
<td>0.38</td>
<td>25927</td>
<td>13</td>
<td>com.compuserve</td>
</tr>
<tr>
<td>0.10</td>
<td>0.03</td>
<td>2058</td>
<td>1</td>
<td>com.yahoo</td>
</tr>
</tbody>
</table>

### Total Transfers from each Archive Section

<table>
<thead>
<tr>
<th>%Reqs</th>
<th>%Byte</th>
<th>Bytes Sent</th>
<th>Requests</th>
<th>Archive Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.07</td>
<td>1.01</td>
<td>68515</td>
<td>52</td>
<td>/junk/</td>
</tr>
<tr>
<td>0.88</td>
<td>0.12</td>
<td>8255</td>
<td>9</td>
<td>/junk/cgi-bin/makeguest.cgi</td>
</tr>
<tr>
<td>0.49</td>
<td>0.07</td>
<td>4645</td>
<td>5</td>
<td>/junk/cgi-bin/tryit.cgi</td>
</tr>
<tr>
<td>0.49</td>
<td>0.03</td>
<td>2015</td>
<td>5</td>
<td>/junk/guest.html</td>
</tr>
<tr>
<td>3.90</td>
<td>0.98</td>
<td>66881</td>
<td>40</td>
<td>/junk/home.html</td>
</tr>
<tr>
<td>0.10</td>
<td>0.08</td>
<td>5575</td>
<td>1</td>
<td>/junk/images/RV.gif</td>
</tr>
<tr>
<td>0.49</td>
<td>0.09</td>
<td>6125</td>
<td>5</td>
<td>/junk/images/archiv.gif</td>
</tr>
<tr>
<td>1.75</td>
<td>0.57</td>
<td>38664</td>
<td>18</td>
<td>/junk/images/gray_bar.gif</td>
</tr>
<tr>
<td>1.85</td>
<td>0.59</td>
<td>40128</td>
<td>19</td>
<td>/junk/images/johnny.gif</td>
</tr>
<tr>
<td>1.75</td>
<td>0.33</td>
<td>22338</td>
<td>18</td>
<td>/junk/images/stats.gif</td>
</tr>
<tr>
<td>1.75</td>
<td>0.43</td>
<td>29036</td>
<td>18</td>
<td>/junk/images/user_man.gif</td>
</tr>
<tr>
<td>1.36</td>
<td>0.32</td>
<td>21700</td>
<td>14</td>
<td>/junk/images/world.gif</td>
</tr>
<tr>
<td>0.10</td>
<td>0.01</td>
<td>752</td>
<td>1</td>
<td>/junk/index.html.bak</td>
</tr>
</tbody>
</table>
This summary was generated by wwwstat-1.0
WebJUNK Server Usage

This is the usage prefix.

Weeks beginning...

Sun Oct 8 00:00:00 1995
Sun Oct 15 00:00:00 1995
Sun Oct 22 00:00:00 1995

Usage by Week

Weeks

Accesses
Name: "one"
Email address: "mmceniry@cs.uah.edu"
Postal address: "three"
Phone number: "four"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "five"

Name: "Michael F. McEniry"
Email address: "mmceniry@cs.uah.edu"
Postal address: "1234 Bob Dr"
Phone number: "205.555.5555"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "You guys are cool."

Name: "Michael F. McEniry"
Email address: "mmceniry@cs.uah.edu"
Postal address: "1234 Bob Dr"
Phone number: "205.555.5555"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "You guys are cool."

Name: "Michael F. McEniry"
Email address: "mmceniry@cs.uah.edu"
Postal address: "1234 Bob Dr"
Phone number: "205.555.5555"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "You guys are cool."

Name: "Michael F. McEniry"
Email address: "mmceniry@cs.uah.edu"
Postal address: "1234 Bob Dr"
Phone number: "205.555.5555"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "You guys are cool."

Name: "Michael F. McEniry"
Email address: "mmceniry@cs.uah.edu"
Postal address: "1234 Bob Dr"
Phone number: "205.555.5555"
Products requested: prod6, prod7, prod8, prod9, prod10
Comments: "You guys are cool."

Name: "Johnny Quest"
Email address: "none@of.your.business"
Postal address: "1234 a street"
Phone number: "976-555-6666"
Products requested: prod3, prod5, prod9
Comments: ""

Name: "James R. Hoffa"
Email address: "jrhoff@davey.jones.com"
The WWW site Manual exists to aid the employee in his or her interaction with the Web JUNK Server. There are areas on Getting Started, which cover the registration of Web pages and the announcement of the pages. Next follow sections on General HTML Authoring, such as creating HTML Documents and creating form documents. Finally, Expansion issues, such as changing, adding, and deleting pages, are covered.

Being a small mom and pop organization, Johnny has not yet needed an official employee handbook. However, we anticipate a huge growth rate for Johnny and have provided this area to meet those expansion needs. Most likely to be included here is information on employee commission rates, health plan, 401K retirement plan, and other employee benefits.

SLAM! Software Solutions
Email: SlamSupport@www.slam.com
INTRODUCTION

This online source is here to aid the employee in his or her interaction with the WebJUNK Server. There are areas on general HTML authoring, advanced HTML applications, such as cgi as used in this project, expandability in the RV product area, and areas such as advertising in general on the WWW.

GETTING STARTED

ANNOUNCING YOUR PAGES
Yahoo does it best! Just look at the tons of services available to help you tell the world that you have arrived!

REGISTERING YOUR PAGES
It's official! A link to InterNIC for registration of your Domain Name.

GENERAL HTML AUTHORTING

CREATING HTML DOCUMENTS
This area provides general information on the basic concepts of HTML.

ADVANCED HTML ISSUES
This area provides general information on the advance topics of HTML used in this project. CGI Form use is highlighted.

EXPANSION ISSUES

CHANGING, ADDING, and DELETING, PAGES
This area provides guidance on making changes to the information in a Web JUNK page (home, product index, product).
APPENDIX D: WAY-FM Web Pages

Stacy Lukins
Christian HIT Radio

‡WAY-FM‡

Where the Music says it all!

Check us out! (Real Time Audio feed coming soon)

Check out this month's TOP TEN LIST!

Stay up to date with WAY-FM NEWS. Catch the latest on the Power Update!

Don't miss out, see this month's CALENDAR OF EVENTS!

Meet the staff! Our EMPLOYEES are great!

Thanks to all of our financial SUPPORTERS!

Check out some of the other great Christian sites on the Web!

WAYM * P.O. Box 887 * Brentwood, TN 37024
1 - 800 - 226 - WAYM(9296)
615 - 370 - 9296

wayfmradio@AOL.com
TOP Hits!

1. I'll Lead You Home - Michael W. Smith
2. Perfect Harmony - Brent Bourgeois
3. The Ocean - The Choir
4. The Narrow Road - Benjamin
5. Take Control - Code of Ethics
6. Liquid - Jars of Clay
7. Reality - Newsboys
8. Love Song - Third Day
9. Richest One - Resurrection Band
10. Always and Forever - Patty Cabrera

RETURN TO WAY-FM HOME
Top Ten List — Artist Info

Michael W. Smith

_Hometown:_ Somewhere, U.S.A

_Albums:_

- Album1
- Album2
- Album3
- Album4

_Record Label:_ XYZ

_Return to Top10List_
WAY ~ FM News!

What's happening at your FAVORITE radio station!

- Newsletter — Monthly newsletter from WAY ~ FM
- Station Locations — Where can you get our signal? Nashville, TN is our home, but we have repeaters all over the Mid-South! We even have a satellite link!
- History — How did WAY ~ FM get started?
- Prayer Requests — Requests from the Station and Listeners. Send us yours!
- Project Updates — Find out about the Power Increase and more!

RETURN TO WAY-FM HOME.
WAY ~ FM Newsletter

WAY ~ FM's newsletter helps keep you abreast of events, staff highlights, letters from listeners, and a monthly Top Ten list!

If you would like to be on the WAY ~ FM postal mailing list to receive your very own copy of this fantastic publication send your mailing address to Judy Edwards.

Current Month (April)

Past Issues

- March '96
- February '96
- January '96
- December '95
- November '95
- October '95
- September '95
- August '95
- July '95
- June '95
- May '95
- April '95

April Newsletter

- Manager's Memo – Power Update
- Spring Shareathon!
- Event Highlights
- Mail Bag

Manager's Memo

To everyone in Middle Tennessee I know you are anxious to see the 88.7 Power Increase become a reality. Believe me, we all have been meeting regularly in prayer huddles lifting this one up. the many delays have been frustrating. However, we are now seeing some real evidence of progress being made. On our front page this month, we are featuring some recent photos taken on the site where the new tower is going.

All of us here are so grateful for the tremendous attitude being displayed by every single WAY listener. You have been totally understanding and patient. Thanks again.

This month, as you can see, work has begun on the new tower. We will keep you posted on when the Power switch to 16,000 watts will actually be occurring.

Your support this month is greatly appreciated and is helping us to see lives changed.

/s/

04/22/96 01:48:35
Spring Shareathon!

PRAY, VOLUNTEER AND LET'S REACH MORE OF THE MID-SOUTH THIS SUMMER AND FALL.

WAY ~ FM stays on the air because of the faithful support of our listeners. Your pledge makes a difference. You can either fill out your pledge card early, or call us during Shareathon itself. (Note: A pre-pledge card will be sent to you in mid April.)

Event Highlights

The D.C. Talk concert with Audio Adrenaline was a huge celebration in downtown Nashville on March 7th. The concert will be repeated on April 11th in Chattanooga at Memorial Auditorium. WAY ~ FM's Michael D'Angelo will be the EmCee. Also, don't miss Michael W. Smith, Jars of Clay, and Three Crosses in Nashville, on April 26th.

WAY ~ FM's outreach to students continues to grow strong. Each Friday night Michael D'Angelo has a group of 10-12 youth in the studio. They are from various youth groups across the Mid-South. Than, the Street Parts kicks into high gear with hip hop jams for Christ. Friday nights are truly dedicated to junior and high school students.

Street Party! check it out every Friday Night at 10PM on WAY-FM!

Mail Bag

WAY ~ FM HAS GROWN INTO NEW COMMUNITIES! RESPONSE HAS BEEN TREMENDOUS.

From a Cleveland, TN listener

Have you ever been under water about to pass out due to lack of oxygen? Then, at the last second be able to rise to the top and catch some welcomed fresh air? This is what WAY ~ FM has been to me. It was February 17 when I came across your station for the first time.

Living in Cleveland, Tennessee, we have had our share of Christian radio station. You name it, they have been here. However, your station is really making a difference.

I listen at 91.3, then when I travel to Chattanooga I can lock you in at 101.5 or 104.7. You are the first station that I have ever sent a donation to. That shows how much I believe in what you are doing.

From a Rome, Georgia listener

Thank you very much!!! Your new translator in rome is an answer to prayer. The music is truly "Out of this World".
WAY ~ FM *Station Locations*

WAY ~ FM studios are located in Brentwood, Tennessee, a suburb of Nashville. We can be heard in Nashville and in cities throughout the mid-South, thanks to repeaters and a digital-quality satellite uplink! Check out our stations in Alabama, Georgia, and Tennessee!

Our FM radio signal can be found in the following locations on these frequencies:

**Alabama**
- Florence, 99.5
- Huntsville, 88.7
- Killen, 89.7

**Georgia**
- Rome, 90.3

**Tennessee**
- Chattanooga, 101.5 & 104.7
- Clarksville, 88.1
- Cleveland, 91.3
- Cookville, 90.1
- Donelson, 99.3
- Fayetteville, 95.7
- Lawrenceburg, 105.3
- Nashville, 88.7 (increasing our power from 3,000 to more than 16,000 Watts very soon!)
- Pulaski, 106.7

**New Areas!**

WAY ~ FM has received permits from the FCC to expand our coverage to Kentucky and Dalton, Georgia. Please pray for us as we try to find sponsors in those areas to pay for equipment installation.

We are constantly trying to expand our coverage area. If you would like to hear WAY ~ FM radio in your area, please contact our General Manager, Bob Augsburg.

Back to News

RETURN TO WAY-FM HOME
WAY ~ FM Loves Its EMPLOYEES!
Check Them Out!

- Bob Augsburg * Manager — "Expansion? Did someone say expansion?!"
- Felice Augsburg * Controller
- Kristen Calpino * Keeps the music flowin' all night long!
- Tim Cardascia * Promotions — "Posters! Get your WAY ~ FM posters! ... Check out the billboards!"
- Michael D'Angelo * Evenings with Michael D' — Top Ten and Friday night: Youth's in the studio!
- Judy Edwards * Office Manager — "Can I send you a newsletter?"
- DJ Majik and crew * Friday Night Street Party — Street music with a Christian message!
- Laurie Lynn * Cheery Morning Personality — "I love hot chocolate!"
- Mercy Machine * Cool WAY ~ FM Van!
- Mark Rider * Programming, Morning Show — "Mark Rider and the CCM Radio Countdown!"
- Ruth Rogers * Receptionist — "WAY ~ FM. May I help you?"
- Matt Thomas * Keeps the afternoons hummin'!
- Michael Wilson * Assistant, Programming

RETURN TO WAY-FM HOME
Keep

CHRIST

in all things!

Here are some of the other Christian sites available on the Web:

RETURN TO WAY-FM HOME.