Modeling and Design of a CM System Content Management System

Version 7.3

Copyright and Licensing Statement

Licenses and Source Code

CM System uses Mozilla's JavaScript C API. See ***http://www.mozilla.org/source.html*** for the source code. In addition, see the Mozilla Public License (***http://www.mozilla.org/source.html***).

Netscape Public License

Apache Software License

IBM Public License

Lesser GNU Public License

Other Copyrights

The CM System installation application was developed using InstallShield, which is a licensed and copyrighted by InstallShield Software Corporation.

The Sprinta JDBC driver is licensed and copyrighted by I-NET Software Corporation.

The Sentry Spellingchecker Engine Software Development Kit is licensed and copyrighted by Wintertree Software.

The Java™ 2 Runtime Environment is licensed and copyrighted by Sun Microsystems, Inc.

The Oracle JDBC driver is licensed and copyrighted by Oracle Corporation.

Contents

C H A P T E R 1

About Modeling and Design

The process of modeling and design must occur before you begin to implement your CM System components. During modeling and design you plan the type of pages you want to display on your Web Site, and then decompose the pages into the components that comprise them so you can determine the Content Types, Templates, and Slots you need in your system.

In addition, the modeling and design process involves planning the shared fields, Global Templates, managed navigation, Roles, Workflows, Communities, and Publishing components required for your system.

The end result of modeling and design is a development plan that specifies the details of the CM System objects that will be implemented.

Although remodeling often occurs during the implementation phase, performing the modeling and design process prior to developing your system is crucial. Experience at Percussion Software consistently demonstrates that customers who engage in preplanning of their implementation are able to implement their system more efficiently and in less time than when those customers that begin implementation before mapping out the structure of their system.

CM System Implementation Roadmap

The CM System implementation roadmap follows. You may find that performing some of these steps in a different order better serves the function of your system. You will also find yourself returning to steps that you have already completed because it has become clear that you must revise some of the components that you have designed.

Steps in the implementation roadmap:

1 Model and design your Web Site and the components that will make up your CM System CMS. Create a development plan that implementers can follow when designing these components. Most of the remaining steps instruct you to create the components designed and outlined during this process.

2 Configure the Roles, Communities, Workflows, and users sketched out during modeling and design. As you continue the implementation process, you will see changes that you want to make.

3 Set up the basic framework for your Site Folders and navigation hierarchy. The Site Folder structure may not be established during modeling and design; you may begin to determine it at this time, and will note changes that you want to make as your implementation proceeds.

4 Create your shared fields.

5 Create your Slots.

6 Create your Global template.

7 Create your Content Types, either by modifying existing FastForward Content Types or by creating new ones.

8 Create your local and shared Templates.

9 Completing the set up of your Site Folders and navigation hierarchy.

10 Modify the configuration of your Roles, Communities, Workflows, and users according to any necessary changes that you have noted during implementation.

11 Configure site folder publishing.

12 Deploy your CM System components to your integration environment, and, after testing, to your production environment.

The implementation roadmap will be represented by the following graphic at the section or chapter that begins each step. The road map will indicate which step you have reached in the process.
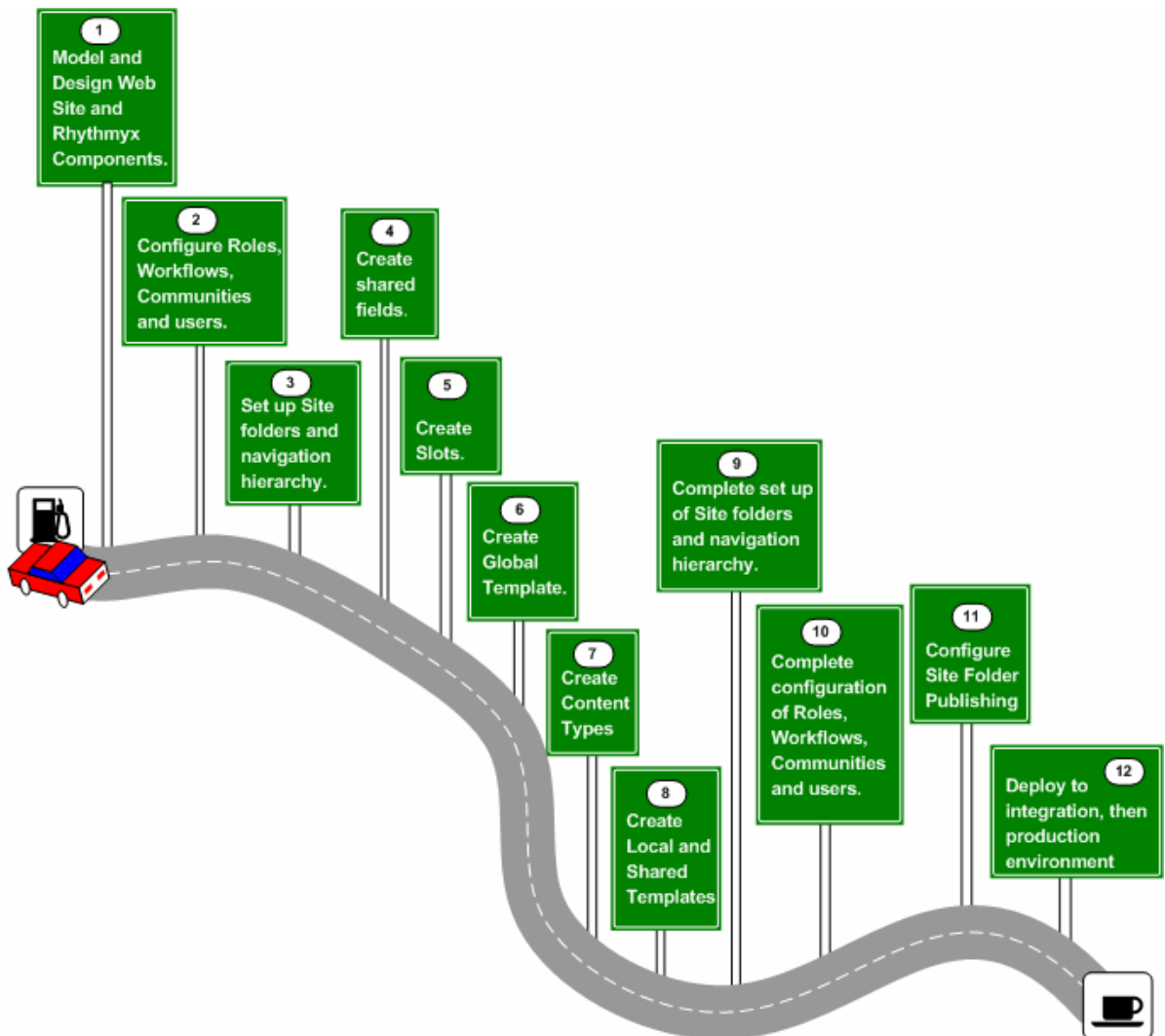
Figure 1: CM System Implementation Roadmap

Modeling and Design in the CM System Implementation Roadmap

This document guides you through the modeling and design process. It is part of the CM System development library, including:

- *Getting Started with CM System*
- *CM System Implementation Guide*
- *Setting Up the CM System Production Environment*

A variety of documents is also available addressing specialized implementation issues.

Modeling and design should precede any implementation activity, but should not occur until after you have developed an initial familiarity with CM System.

Before beginning modeling and design, you should become familiar with CM System and its basic concepts.

- Read the *CM System Concepts Guide*.

  This document introduces and explains the basic concepts of CM System and of Content Management using CM System. You should read at least the portions of the *CM System Concepts Guide* recommended for implementers.

- Read *Getting Started with CM System*.

  This document guides you through a basic installation of CM System with the FastForward implementation and includes some basic tutorial exercises to help you learn more about CM System and how it works.

- Attend training on CM System.

  Percussion Software provides training on CM System frequently throughout the year. Training will provide more opportunities to become familiar with CM System and the implementation process.

Once you have completed these tasks, you are ready to use this document to help guide you through the modeling and design process.

C H A P T E R 2

Preparing for Modeling and Design

Just as performing modeling and design will result in an improved implementation, preparing for modeling and design ensures that that process will be completed quickly and effectively. Review the modeling and design process before beginning. If Percussion Software is assisting with your modeling and design, you will be provided with a Pre-Engagement Workbook. Complete this document before beginning your modeling and design session with Percussion Professional Services Organization representatives.

It is also useful to provide printed copies of the HTML pages you want to publish. If you are redesigning your pages, you can use printouts of the design wireframes. Otherwise, use printouts of pages from your existing Web site.

Steps in the Modeling and Design Process

This topic provides a set of steps that may be followed during modeling and design. However, many factors may change the order in which you perform the steps including the number of pages on your Web Site and the degree to which you are changing your current Web Site.
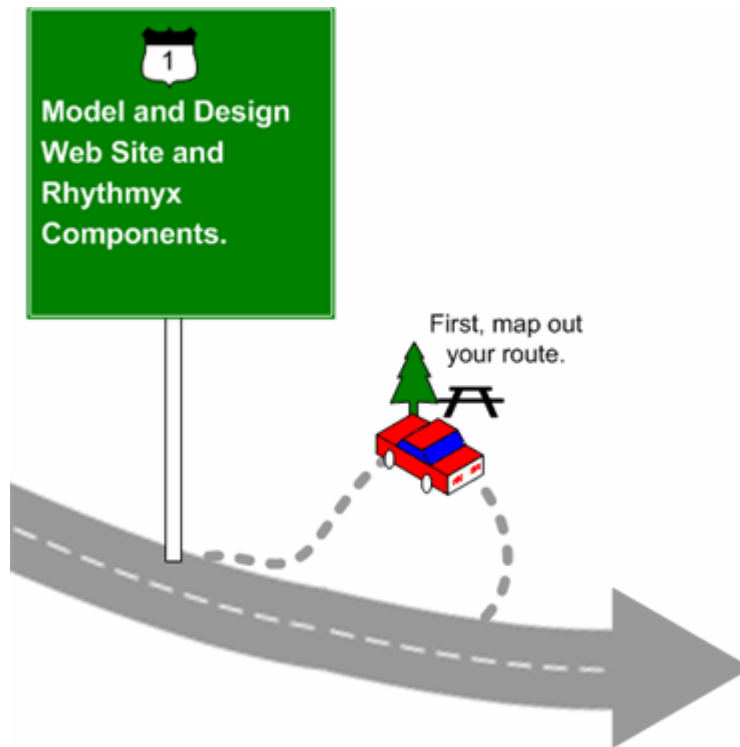
Steps for modeling and design:

1 *Map out your Site and its changes* (see page 14).

2 *Perform an initial assessment of the Content Types required on your Site.* (see page 20)

3 *Analyze Content Type fields and related Content Items used on Web pages.* (see page 27)

4 *Decompose pages into fields, Templates, and Slots.* (see page 35)

5 *Refine and list fields required on Content Types.* (see page 47)

6 *Design Global Template and Managed Navigation.* (see page 85)

7 *Plan Roles, Communities and Workflows.* (see page 95)

8 *Determine publishing requirements.* (see page 109)

9 *Produce a development plan.* (see page 117)

Our Modeling and Design Sample

To demonstrate modeling and design we will reproduce a session between a customer who will be involved in implementing the system and a CM System implementer. In this exercise, the customer's Site will be represented by a small portion of the FastForward Enterprise Investments Site. Using a small portion of the Site will enable us to demonstrate how analysis occurs and how decisions are made while avoiding the length of discussion necessary for analyzing an entire Site. The portion of the Site that we use will include many of the Content Types, Templates, Slots, shared fields and other components that we implement in Section 2 of this document.

At the beginning of each topic in this section, we include a box that summarizes the main question or questions that the CM System implementer is asking. The section shows how the customer answers or responds to the question or questions.

Mapping Out The Site

Questions:
- What do you want your CM System-designed Web Site to look like?
- With your implementation of CM System will you be duplicating your current site, redesigning your site, or designing your site for the first time?

The process of mapping out a Site varies greatly from customer to customer. In most cases, you will begin with an existing Web Site and include any changes that you want to make during your transition to CM System. If you do not have an existing Web Site, you must sketch out your proposed Web Site.

The complexity of this step is affected by the following:

- the size of the Site
- the number of sub-sections
- the number of page types
- connections to outside systems

During an actual modeling and design session, all sub-sections, page types, and connections to other systems must be mapped out on paper.

For demonstration purposes, we will use a very small portion of the FastForward Site to represent the Site of our customer, Ally.

The graphic that follows shows the portion of Enterprise Investments that we will look at while analyzing Ally's Site. Ally would print out or sketch the pages much as they appear in the graphic, and show how they are connected with arrows.

Note that in the FastForward Enterprise Investments Site, the Home Page does not directly link to the "Better Investing National Convention and World Federation of Investors" page. However, in this document, we do not discuss the December 2004 Calendar Page, which links these two pages. Therefore, it is only represented by a placeholder.

As Ally and Noah, the CM System implementer who is working with her, proceed to analyze these pages in the topics that follow, you will have an opportunity to see larger graphics of them. The graphic in the next page shows you the pages comprising Ally's Web Site and shows how they are connected, but does not clearly display any page details.

Our purpose for using the Home Page in this exercise is to maintain the starting point of the Site, and to discuss how its graphics are stored in Content Types. We will not decompose the Home Page as we decompose the other Pages because it uses custom features which are beyond the scope of this document.
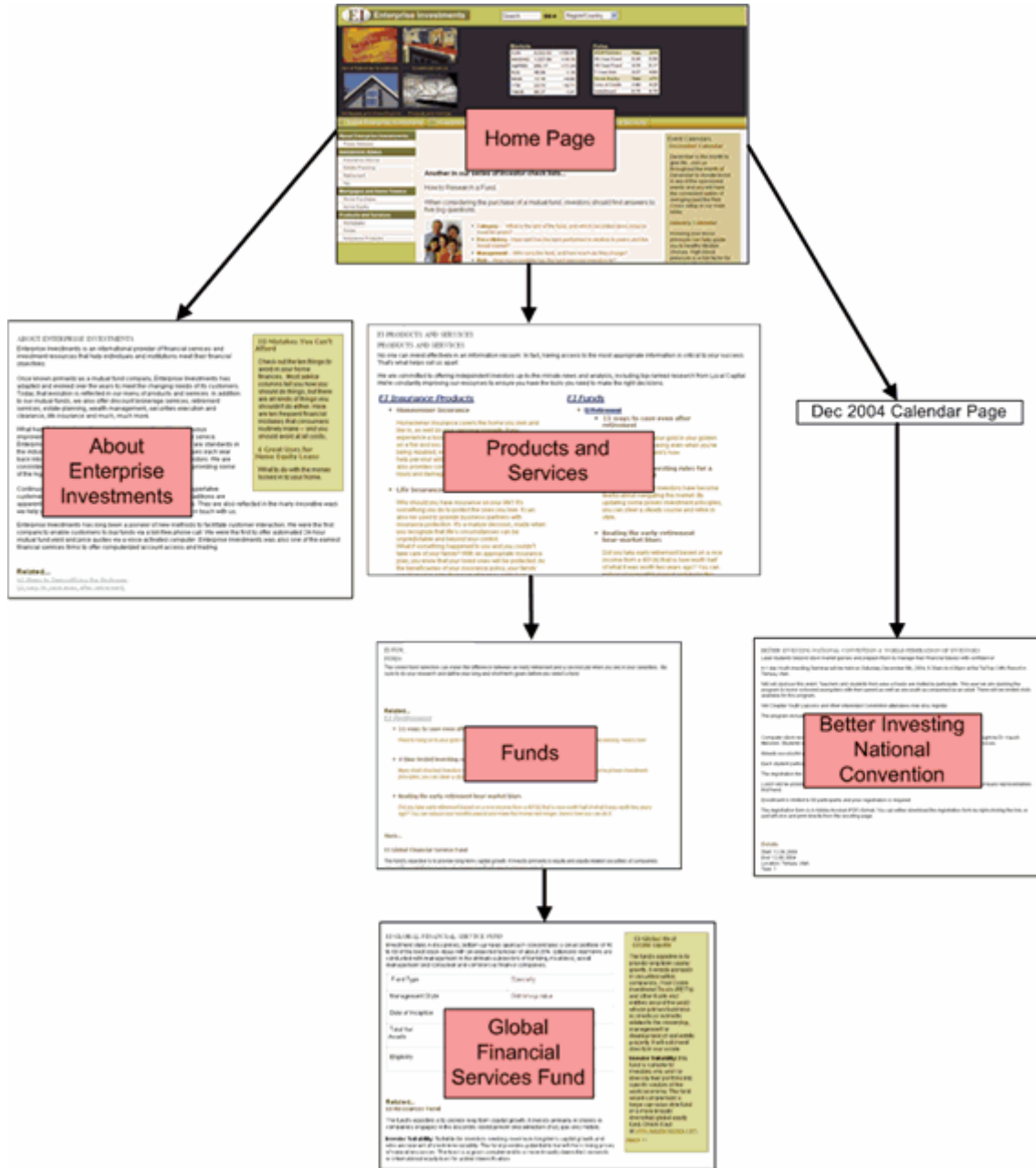
Figure 2: Ally's Sample Site

Revisions from the Original Site

Question:
- What aspects of your current Site do you want to redesign with CM System?

Although you may have several changes that you want to make to the pages in your Site when you move to CM System, in our example, Ally plans to make one change to the "Better Investing Convention . . . " Page. She shows the change she wants to make to Noah to discuss its feasibility. The current "Better Investing Convention . . . " Page appears as:

BETTER INVESTING NATIONAL CONVENTION & WORLD FEDERATION OF INVESTORS
Lead students beyond stock market games and prepare them to manage their financial futures with confidence!

A 1-day Youth Investing Seminar will be held on Saturday, December 6th, 2004, 8:30am to 4:00pm at the TipTop Cliffs Resort in Tempa, Utah.

NAI will sponsor this event. Teachers and students from area schools are invited to participate. This year we are opening the program to home schooled youngsters with their parent as well as any youth accompanied by an adult. There will be limited slots available for this program.

NAI Chapter Youth Liaisons and other interested Convention attendees may also register.

The program includes an introduction to NAI-style investing, through group simulation activities.

Computer stock research techniques will be taught, using NAI software. Quick Analysis of stocks will be taught by Dr. Kaush Meisheri. Students will discover investing basics in planning their financial futures and in making career choices.

Already successful young investors will be on hand to assist NAI teachers.

Each student participant receives a copy of the NAI's Investing for Life Youth Handbook.

The registration fee is $25 per adult. This fee allows each adult to bring up to three students.

Lunch will be provided. At the end of the program, students will visit the company exhibitors and talk with company representatives first hand.

Enrollment is limited to 50 participants and prior registration is required.

The registration form is in Adobe Acrobat (PDF) format. You can either download the registration form by right-clicking the link, or just left-click and print directly from the resulting page.

Details

Start: 12.06.2004
End: 12.06.2004
Location: Tempa, Utah
Type: 1

Figure 3: Original Better Investing National Convention Page

Events at Ally's company occur simultaneously at various Locations. In the past, content contributors have had to display a separate page on the Web Site for the same event at each location. She would like the ability to create one Content Item that stores the event and its multiple locations, and the ability to display the event and all of the locations on one Web page. Ideally, she would like each Location entry to hold four fields describing the Location: Address, City, State, and Contact. The final page would appear something like the following, where the additional locations are listed at the bottom of the page.

BETTER INVESTING NATIONAL CONVENTION & WORLD FEDERATION OF INVESTORS
Lead students beyond stock market games and prepare them to manage their financial futures with confidence!

A 1-day Youth Investing Seminar will be held on Saturday, December 6th, 2004, 8:30am to 4:00pm at the TipTop Cliffs Resort in Tempa, Utah.

NAI will sponsor this event. Teachers and students from area schools are invited to participate. This year we are opening the program to home schooled youngsters with their parent as well as any youth accompanied by an adult. There will be limited slots available for this program.

NAI Chapter Youth Liaisons and other interested Convention attendees may also register.

The program includes an introduction to NAI-style investing, through group simulation activities.

Computer stock research techniques will be taught, using NAI software. Quick Analysis of stocks will be taught by Dr. Kaush Meisheri. Students will discover investing basics in planning their financial futures and in making career choices.

Already successful young investors will be on hand to assist NAI teachers.

Each student participant receives a copy of the NAI's Investing for Life Youth Handbook.

The registration fee is $25 per adult. This fee allows each adult to bring up to three students.

Lunch will be provided. At the end of the program, students will visit the company exhibitors and talk with company representatives first hand.

Enrollment is limited to 50 participants and prior registration is required.

The registration form is in Adobe Acrobat (PDF) format. You can either download the registration form by right-clicking the link, or just left-click and print directly from the resulting page.

Details
Start: 12.06.2004
End: 12.06.2004
Type: 1

Locations
6 Sparrow Way, Tempa, Utah contact: Louise Ng
131 West 4th Street, New York, NY contact: Lisa Shue
2043 Pine Street, San Francisco, CA contact: Gil Bronstein

Figure 4: New Page for Better Investing National Convention

Noah says that CM System can handle this change easily. He proposes a method for doing this later, in the topic ***Decomposing the Events Page*** .

In Ally's mock-up of her proposed Web Site, she replaces the original "Better Investing National Convention" page with the modified one.

C H A P T E R 3

Initial Evaluation of the Web Site

The first phase of modeling and design is to perform an initial evaluation of the Web site to outline the basic Content Types that will be used to define managed content, and preliminary Templates that will be used to publish them.

Initial Assessment of Content Types

Question: ▪   What are your initial ideas of the Content Types that your system will require?

Noah suggests that Ally begin by making initial assessments of what Content Types her different pages should represent. He explains that nothing decided at this point will be final, and that she should not feel pressured to make the most efficient decisions.

As stated earlier, we will not decompose the Home Page like the other Pages in Ally's site due to its custom components. We will discuss this page last, so that we can address the use of image Content Types.

Ally begins with the other pages in her Site, and Noah explains that the uniform area surrounding each page is a Global Template, which they will analyze later, and not include in their analysis of the Pages. He instructs her to cut it out (as we have done in our diagram) or ignore it as she analyzes her pages.

The grayed out portion in the following graphic marks off the Global Template.

Figure 5: Global Template

NOTE: In the graphic above, breadcrumbs are part of the global template, as they normally will be. In FastForward, due to the HTML requirements of the structure of the page, breadcrumbs are part of each main page template rather than the global template.

Ally and Noah begin by looking at the About Enterprise Investments page.

Figure 6: About Enterprise Investments

Ally notes that it includes a title and content describing generic information about the company. Ally suggests that it represents a Generic Content Type. Noah feels that this is a good initial assessment, but mentions that Ally should make a note of the links to related pages at the bottom of the page and the box on the side including title links and descriptions to other related pages. He explains that these elements may help Ally determine which other pages can use the same Content Type as this one.

Next they look at the Products and Services page:

Figure 7: Products and Services page

Noah asks Ally to describe the page, and then make an initial assessment of what Content Type it represents. She notes that it has the Products and Services title, a brief introduction, and then a section with two headings for categories of products: Insurance Products and Funds. In this section:

    1   the headings are links to articles about the category of product

    2   below the headings are title links to specific types of insurance or funds followed by detailed body text.

Ally says that her first impression is that this represents a Products and Services Content Type since all of the content is related to Products and Services. Noah has her temporarily label the page "Products and Services Content Type", but suggests that this assessment may change as she looks at her other pages.

Now Ally and Noah look at the Funds Page:

EI FUNDS

FUNDS

The correct fund selection can mean the difference between an early retirement and a second job when you are in your seventies. Be sure to do your research and define your long and short-term goals before you select a fund.

Related...

EI Retirement

- **11 ways to save even after retirement**

  Want to hang on to your gold in your golden years? Then keep saving even when you've stopped earning. Here's how.

- **4 time-tested investing rules for a scary new world**

  Many shell-shocked investors have become fearful about navigating the market. By updating some proven investment principles, you can steer a steady course and retire in style.

- **Beating the early-retirement bear-market blues**

  Did you take early retirement based on a nice income from a 401(k) that is now worth half of what it was worth two years ago? You can reduce your monthly payout and make the money last longer. Here's how you can do it.

More...

EI Global Financial Service Fund

The fund's objective is to provide long-term capital growth. It invests primarily in equity and equity-related securities of companies around the world that are primarily engaged in the financial services industry.

Investor suitability: Suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. This fund would complement a large-cap value style fund or a more broadly diversified global equity fund.

EI Global Health Sciences

The fund's objective is to provide long-term capital growth. It invests primarily in equity and equity-related securities of companies around the world that are engaged in the development, production or distribution of products and services related to healthcare.

Investor Suitability: This fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund.

EI Global Real Estate Equity

The fund's objective is to provide long-term capital growth. It invests primarily in securities within companies, Real Estate Investment Trusts (REITs) and other trusts and entities around the world whose primary business is directly or indirectly related to the ownership, management or development of real estate property. It will not invest directly in real estate.

Investor Suitability: this fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund. Check it out at www.investersplace.com .

EI Resources Fund

The fund's objective is to provide long-term capital growth. It invests primarily in shares in companies engaged in the discovery, development and extraction of oil, gas and metals.

Investor Suitability: Suitable for investors seeking maximum long-term capital growth and who are tolerant of short-term volatility. The fund provides potential to benefit from rising prices of natural resources. The fund is a good complement to a more broadly diversified domestic or international equity fund for added diversification.

Figure 8: Funds Page

Ally notes that the Funds page begins with the Funds title and a brief introduction, similar to the Products and Services page. Then it includes a Related section that resembles the second column of the Products and Services page. The column includes the same list of retirement articles and funds listed in the second column of the Products and Services page as well as two additional funds. Since the page is dedicated to information about Funds, Ally decides to call it a Funds Page. Noah agrees that she should temporarily label the page "Funds Page", but asks her to keep in mind that this page and the previous one have the following similarities in format:

- At the top, the title is followed by an introduction.
- After the title and introduction is a section with links to articles followed by links to specific products with lengthy descriptions of them. The only differences is that one page has a two-column format, and the other page has a single column format.

Next they look at the EI Global Financial Service Fund page:

EI GLOBAL FINANCIAL SERVICE FUND
Investment style: A disciplined, bottom-up value approach concentrates a small portfolio of 40 to 50 of the best stock ideas with an expected turnover of about 25%. Extensive interviews are conducted with management in the primary subsectors of banking, insurance, asset management and consumer and commercial finance companies.

Fund Type	Specialty
Management Style	Bottom-up value
Date of Inception	April 2000
Total Net Assets	$16.8 million
Eligibility	RESP, foreign content portion of RSP & RIF

Related...
EI Resources Fund

The fund's objective is to provide long-term capital growth. It invests primarily in shares in companies engaged in the discovery, development and extraction of oil, gas and metals.

Investor Suitability: Suitable for investors seeking maximum long-term capital growth and who are tolerant of short-term volatility. The fund provides potential to benefit from rising prices of natural resources. The fund is a good complement to a more broadly diversified domestic or international equity fund for added diversification.

EI Global Real Estate Equity

The fund's objective is to provide long-term capital growth. It invests primarily in securities within companies, Real Estate Investment Trusts (REITs) and other trusts and entities around the world whose primary business is directly or indirectly related to the ownership, management or development of real estate property. It will not invest directly in real estate.

Investor Suitability: this fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund. Check it out at www.investersplace.com.

more >>

Figure 9: EI Global Financial Service Fund page

Ally points out that this page includes the title of the specific type of fund and a description as well as a table giving specific items of information about the fund. At the bottom of the page is a related fund in the format of a title link and description, and at the right side of the page in a box is another fund with a title link and description. Ally designates this page as a representing a "Fund Type" Content Type. Noah suggests that the page has quite a few similarities to the "About Enterprise Investments" page, but suggests that they investigate this after looking at the other pages in the Site.

The next page that they look at is the "Better Investing National Convention" page that Ally is planning to change:

BETTER INVESTING NATIONAL CONVENTION & WORLD FEDERATION OF INVESTORS
Lead students beyond stock market games and prepare them to manage their financial futures with confidence!

A 1-day Youth Investing Seminar will be held on Saturday, December 6th, 2004, 8:30am to 4:00pm at the TipTop Cliffs Resort in Tempa, Utah.

NAI will sponsor this event. Teachers and students from area schools are invited to participate. This year we are opening the program to home schooled youngsters with their parent as well as any youth accompanied by an adult. There will be limited slots available for this program.

NAI Chapter Youth Liaisons and other interested Convention attendees may also register.

The program includes an introduction to NAI-style investing, through group simulation activities.

Computer stock research techniques will be taught, using NAI software. Quick Analysis of stocks will be taught by Dr. Kaush Meisheri. Students will discover investing basics in planning their financial futures and in making career choices.

Already successful young investors will be on hand to assist NAI teachers.

Each student participant receives a copy of the NAI's Investing for Life Youth Handbook.

The registration fee is $25 per adult. This fee allows each adult to bring up to three students.

Lunch will be provided. At the end of the program, students will visit the company exhibitors and talk with company representatives first hand.

Enrollment is limited to 50 participants and prior registration is required.

The registration form is in Adobe Acrobat (PDF) format. You can either download the registration form by right-clicking the link, or just left-click and print directly from the resulting page.

Details
Start: 12.06.2004
End: 12.06.2004
Type: 1

Locations
6 Sparrow Way, Tempa, Utah contact: Louise Ng
131 West 4th Street, New York, NY contact: Lisa Shue
2043 Pine Street, San Francisco, CA contact: Gil Bronstein

Figure 10: New Page for Better Investing National Convention

Ally notes that like the other pages they have looked at so far, this page has a title and a description - enough to compose the body of the Content Type. It differs from the other pages reviewed so far because at the bottom of the page it lists certain event details - the event start date, event end date, event type, and event locations, rather than links to related pages. Ally designates this as an Event Content Type. Because of the event-specific information listed, Noah agrees that this may work as an Event Content Type, but he reiterates that nothing is final at this point.

Finally, Noah and Ally look at the home page. Their complete analysis of this page will not be covered in this document. However, we will cover their discussion of the various images shown on this page:

Figure 11: Home Page

Noah explains that in CM System, images and other binary files must be stored in a Content Type that includes a special control for uploading and storing binary files as well as certain metadata fields that are required for storing, managing, and rendering binary files. Image files need additional metadata fields required for rendering them on a Web page. Ally tells Noah that she was hoping to have the flexibility of an image Content Type that could create a thumbnail of an uploaded image. Noah explains that this is frequently requested, and they can include a java extension (plugin) with the Content Type to create the thumbnails. Ally designates an Image Content Type to store images of the type she sees on the Home page and thumbnails of the image.

Noah explains that Ally is now ready to begin decomposing the pages she has temporarily designated as various Content Types into their component fields to make a better determination about which of the pages are similar and which are different.

Initial Web Page Decomposition

Questions:
- What is similar and different about the pages you have sketched out?
- Which of these pages potentially could be represented by the same Content Types?

At this point, Ally has loosely defined the Content Types that her Web pages represent as Generic, Products and Services, Funds, Fund Type, Event, and Image. Now, Noah explains, they will roughly decompose the pages into fields and links to better see their structural similarities. This process may help them see that pages that initially appear different because they deal with different topics may actually be similar in structure. If pages are similar in structure, Ally can use the same Content Type to define their content.

Noah suggests beginning by comparing the "Funds" page and the "Generic" pages:

EI FUNDS

FUNDS

The correct fund selection can mean the difference between an early retirement and a second job when you are in your seventies. Be sure to do your research and define your long and short-term goals before you select a fund.

Related...

EI Retirement

- **11 ways to save even after retirement**

  Want to hang on to your gold in your golden years? Then keep saving even when you've stopped earning. Here's how.

- **4 time-tested investing rules for a scary new world**

  Many shell-shocked investors have become fearful about navigating the market. By updating some proven investment principles, you can steer a steady course and retire in style.

- **Beating the early-retirement bear-market blues**

  Did you take early retirement based on a nice income from a 401(k) that is now worth half of what it was worth two years ago? You can reduce your monthly payout and make the money last longer. Here's how you can do it.

More...

EI Global Financial Service Fund

The fund's objective is to provide long-term capital growth. It invests primarily in equity and equity-related securities of companies around the world that are primarily engaged in the financial services industry.

Investor suitability: Suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. This fund would complement a large-cap value style fund or a more broadly diversified global equity fund.

EI Global Health Sciences

The fund's objective is to provide long-term capital growth. It invests primarily in equity and equity-related securities of companies around the world that are engaged in the development, production or distribution of products and services related to healthcare.

Investor Suitability: This fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund.

EI Global Real Estate Equity

The fund's objective is to provide long-term capital growth. It invests primarily in securities within companies, Real Estate Investment Trusts (REITs) and other trusts and entities around the world whose primary business is directly or indirectly related to the ownership, management or development of real estate property. It will not invest directly in real estate.

Investor Suitability: this fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund. Check it out at www.investersplace.com .

EI Resources Fund

The fund's objective is to provide long-term capital growth. It invests primarily in shares in companies engaged in the discovery, development and extraction of oil, gas and metals.

Investor Suitability: Suitable for investors seeking maximum long-term capital growth and who are tolerant of short-term volatility. The fund provides potential to benefit from rising prices of natural resources. The fund is a good complement to a more broadly diversified domestic or international equity fund for added diversification.

Figure 12: Funds Page

Figure 13: About Enterprise Investments

Noah asks Ally to note the similarities and differences between these two pages. She notes that they both have a title at the top and then body text. Following the body text they both have a Related section. She states that the Related sections are quite different. The Funds page includes title links and brief summaries for retirement articles, and then title links and more substantial descriptions for funds. The Related section for the About Enterprise Investments page includes two simple title links to other pages. At the side of the page is a box with title links and descriptive text for two additional pages.

Noah explains which parts of the pages would represent fields in the Content Types in CM System, and which parts of the pages would represent information from other pages and would not be part of the Content Type. He explains that the titles and body text are part of the Content Types and would probably be represented by "Title" and "Body" fields. The remainder of the information on both of the pages is information from other Content Items that links to pages representing those Content Items.

Ally sees that structurally, these pages are more similar than she realized. They both only display a title and body field from the original Content Item. All of the other information they display includes text and links to other Content Items, although much of this information appears in different places on the page.

Noah explains that they are not actually looking at Content Types, but *Templates* that display Content Items of certain Content Types in specific formats. He explains that Templates display some fields from a Content Item and can also display portions of other Content Items in *Slots* that are included in Templates. It is possible for Templates to hide the empty space where a field on a Content Type is not filled in; it is also possible for Templates to hide Slots where other Content Items could be displayed, but are currently empty. For example, a Slot on the Funds page might include a green bar on the right side with data from other Content Items, but since the Web Master has not entered other Content Items into that Slot, the green bar is not shown on that page.

Noah suggests that the two pages could actually represent the same Content Type, and could possibly represent the same Template. Ally objects, pointing out that the two pages cannot represent the same Template because, although they both display links to pages for other Content Items at the bottom of the page, the format of the links is completely different. Noah explains that in CM System, the links and portions of other Content Items on pages are also Templates, and that any number of differently formatted Templates can be inserted into a Slot. Therefore, it is possible that the Funds and the About Enterprise Investments pages could represent the same Template and the same Content Type. However, Ally is not sure if she wants the creators of the Funds page to have the option of using the green right sidebar to present additional information. Noah also points out that if it turns out that the different formats of the Content Items in the Slots would require too much additional coding in the Page Template, it may not be worthwhile to use the same Template for both of them, although they could still represent the same Content Type.

Noah confirms that Ally is beginning to think about her Content Types and Templates from a CM System perspective. He says that she should keep in mind the question of whether the two pages looked at so far should represent a single Content Type or two separate Content Types, but can make this decision after they look at the other pages again.

Now they look at the EI Global Financial Service Fund page:

Figure 14: EI Global Financial Service Fund page

Ally states that this page is essentially the same structurally as the About Enterprise Investments page. It has a title and body text. On the right is the green sidebar displaying another Content Item, and at the bottom is a "Related" section displaying another Content Item. Noah points out the table in the center of the page, and says that the Fund Type, Management Style, Date of Inception, Total Net Assets, and Eligibility fields should probably be specific to the Content Type representing the fund.

At this point, Ally gives Noah some additional information. There are actually several funds represented in pages similar to this one on the Web Site. They all have a table with these pieces of information in the center. However, the entire table is copied and pasted from documents sent from another department. The content contributors do not require separate fields for entering the information in the table.

With this information, Noah suggests that the table can be part of the body content which can be entered into a rich text field in the Content Editor in the following format:
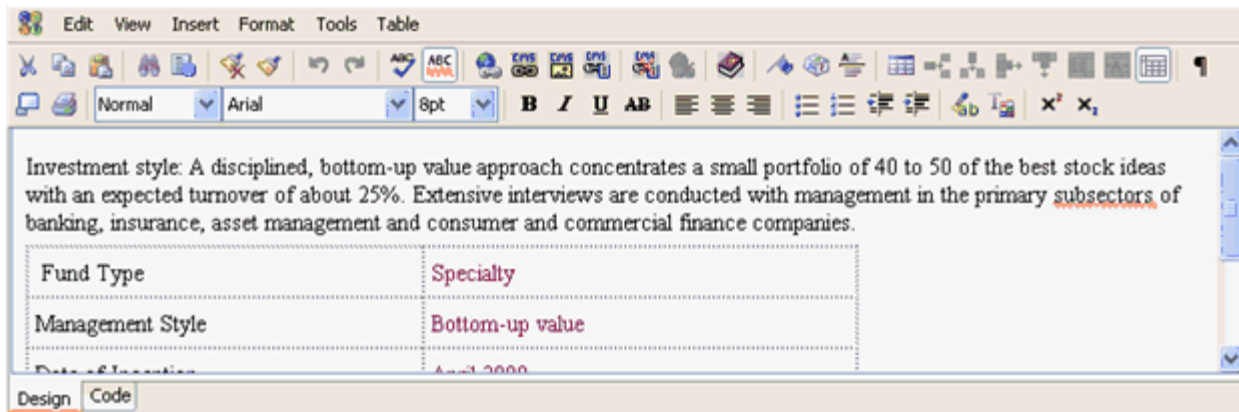
Figure 15: Content entered in rich text control

How to decide whether to include data in its own field or as part of a rich text field:

Include data in its own field if:

- The data includes multiple values that may change more frequently than the content in the rich text field.
- You do not always want the data assembled with the content in the rich text field.
- You want to require users to enter this data for the Content Type.
- You want to specify that the field appear in a particular location on the page (not within the rich text field).

Include data as part of a rich text field if:

- By inserting the data into the rich text field, you are able to avoid creating a new Content Type.
- When assembled, the content in the rich text field should always be accompanied by the data.
- The data is not entered often enough to require its own field or fields.
- You want to give the content contributor control over where the field appears on every page.

Now Noah agrees that the Global Financial Services Fund page does have the same format as the About Enterprise Investments page. Ally decides that her system should include a Generic Content Type that could be used for information as diverse as the Site's summary page, and pages representing specific funds. She notes that it could also be used for "major products" pages such as the Funds page if she decides that the Funds page should have the green sidebar Slot on the right side. Noah reminds her that the sidebar Slot is part of the Template, not the Content Type. So Ally decides that a Generic Content Type should represent all three pages decomposed so far, the About Enterprise Investments page, the EI Global Financial Service Fund page, and the Funds page. The main fields on each of theses page seems to be the title and body text. She notes that different Page Templates may be associated with these pages.

They now look at the Products and Services page again.

PRODUCTS AND SERVICES

No one can invest effectively in an information vacuum. In fact, having access to the most appropriate information is critical to your success. That's what helps set us apart.

We are committed to offering independent investors up-to-the-minute news and analysis, including top-ranked research from Local Capital. We're constantly improving our resources to ensure you have the tools you need to make the right decisions.

EI Insurance Products

- **Homeowner Insurance**

    Homeowner insurance covers the home you own and live in, as well as your personal property. If you experience a loss, which is covered by insurance, such as a fire and you cannot live in your home while it is being repaired, we'll provide you with a benefit that will help pay your additional living expenses. This insurance also provides coverage for your personal liability for injury and damage.

- **Life Insurance**

    Why should you have insurance on your life? It's something you do to protect the ones you love. It can also be used to provide business partners with insurance protection. It's a mature decision, made when you recognize that life's circumstances can be unpredictable and beyond your control.
    What if something happened to you and you couldn't take care of your family? With an appropriate insurance plan, you know that your loved ones will be protected. As the beneficiaries of your insurance policy, your family would receive a tax-free sum of money, called an insurance benefit. Because you planned ahead, your family would be able to lead a comfortable life and build a secure future.

- **Reinsurance**

EI Funds

- **EI Retirement**
    - **11 ways to save even after retirement**

        Want to hang on to your gold in your golden years? Then keep saving even when you've stopped earning. Here's how.

    - **4 time-tested investing rules for a scary new world**

        Many shell-shocked investors have become fearful about navigating the market. By updating some proven investment principles, you can steer a steady course and retire in style.

    - **Beating the early-retirement bear-market blues**

        Did you take early retirement based on a nice income from a 401(k) that is now worth half of what it was worth two years ago? You can reduce your monthly payout and make the money last longer. Here's how you can do it.

    More...
- **EI Global Financial Service Fund**

Figure 16: Products and Services page

Like the other pages they have looked at so far, it includes title and body fields at the top, which are followed by content from other content items. Potentially, it could have a green box on the side representing a Slot. The only major difference between this page and the other pages is that the other content included at the bottom of this page appears in two columns. Noah mentions that the two (or possibly more) column approach on the Products and Services page may require different coding in the Template than the other pages, but since the fields that come from the Content Type (title and body) are the same, they should be able to use the same Content Type. He continues that since the four pages use the same fields from the original Content Type, the same Content Type can represent all of them, and Ally determines that the Generic Content Type will represent all four of them.

They now look at the Better Investing National Convention page:

BETTER INVESTING NATIONAL CONVENTION & WORLD FEDERATION OF INVESTORS

Lead students beyond stock market games and prepare them to manage their financial futures with confidence!

A 1-day Youth Investing Seminar will be held on Saturday, December 6th, 2004, 8:30am to 4:00pm at the TipTop Cliffs Resort in Tempa, Utah.

NAI will sponsor this event. Teachers and students from area schools are invited to participate. This year we are opening the program to home schooled youngsters with their parent as well as any youth accompanied by an adult. There will be limited slots available for this program.

NAI Chapter Youth Liaisons and other interested Convention attendees may also register.

The program includes an introduction to NAI-style investing, through group simulation activities.

Computer stock research techniques will be taught, using NAI software. Quick Analysis of stocks will be taught by Dr. Kaush Meisheri. Students will discover investing basics in planning their financial futures and in making career choices.

Already successful young investors will be on hand to assist NAI teachers.

Each student participant receives a copy of the NAI's Investing for Life Youth Handbook.

The registration fee is $25 per adult. This fee allows each adult to bring up to three students.

Lunch will be provided. At the end of the program, students will visit the company exhibitors and talk with company representatives first hand.

Enrollment is limited to 50 participants and prior registration is required.

The registration form is in Adobe Acrobat (PDF) format. You can either download the registration form by right-clicking the link, or just left-click and print directly from the resulting page.

Details
Start: 12.06.2004
End: 12.06.2004
Type: 1

Locations
6 Sparrow Way, Tempa, Utah contact: Louise Ng
131 West 4th Street, New York, NY contact: Lisa Shue
2043 Pine Street, San Francisco, CA contact: Gil Bronstein

Figure 17: New Page for Better Investing National Convention

When Ally initially assessed this page, she noted that it showed fields that seemed specific to events, and therefore should represent an Event Content Type. Looking at it again she decides that this assessment is sound. The page includes an Event Start Date, Event End Date, Event Type field, and the Event Location table. If she associated this page with the Generic Content Type, she would have to add all of these fields to the Generic Content Type, even though they would only be used for Events. Associating the page with an Event Content Type is more efficient.

How to decide whether to merge similar pages into one Content Type or split them into two Content Types:

Merge similar pages into one Content Type if:

- You want to give content contributors or web masters a great deal of flexibility regarding the content they enter and how they format it.

- Differences are negligible; for example by adding a date field or a basic Slot to one of the Content Types, you could merge them into one and make your system simpler.

Split similar pages into two Content Types if:

- Merging would require you to add fields to one of the Content Types that you do not want available for that Content Type.

Ally and Noah have already decided that a separate Image Content Type is required, so at this point, they do not have to further analyze the Home Page. Noah suggests that this is a good time to begin thinking about the specifics of the Templates that Ally requires to display her content.

C H A P T E R 4

Designing Templates and Slots

Noah has already introduced the concept of Templates to Ally, but before they begin designing Templates for her system, he explains CM System Templates in more detail. Noah explains that the purpose of Templates is to define how CM System outputs are assembled and formatted. These outputs can be Pages or portions of Pages, as Ally has already seen. For example, a Page Template defines the entire About Enterprise Investments page. In the following graphic of the Template used for this Page, the title and body fields from the Content Item are outlined, and Slots holding Content Items are shaded:
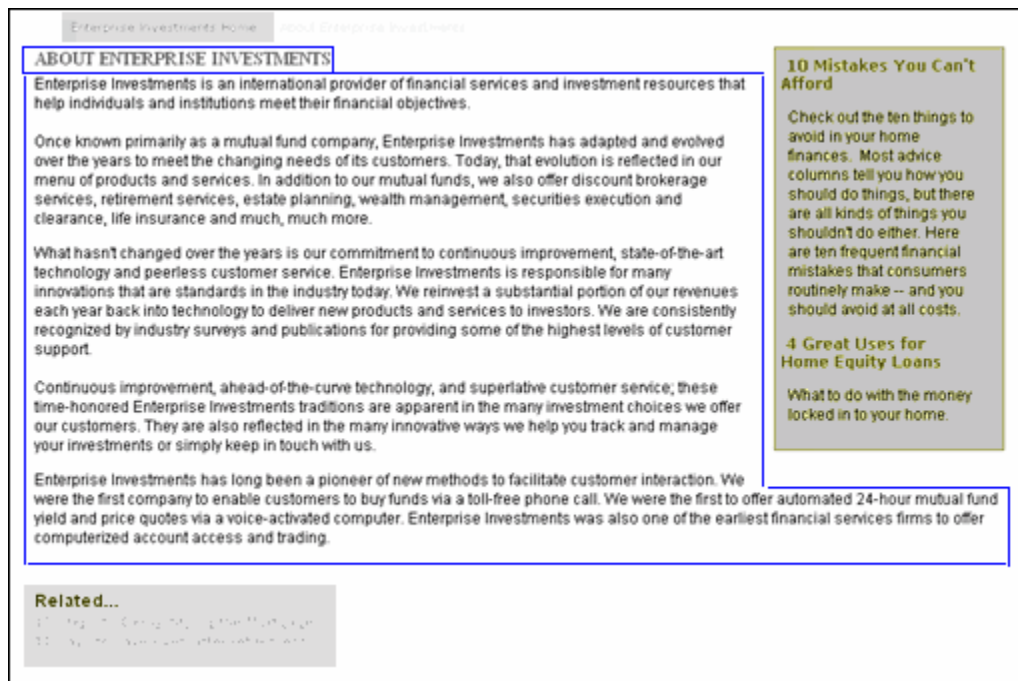
Figure 18: Page Template

When an output cannot stand alone as a Page, in CM System it is referred to as a Snippet. In the above graphic, Snippets appear in placeholders on the Template called *Slots*. A Snippet can contain any fields in a Content Item, including fields that are not included in any Page Templates. For example, a Snippet Template defines the green sidebar on the About Enterprise Investments page. This Snippet can include multiple Content Items. In this case, it includes the title and summary fields from two Content Items, which are outlined in the following graphic:
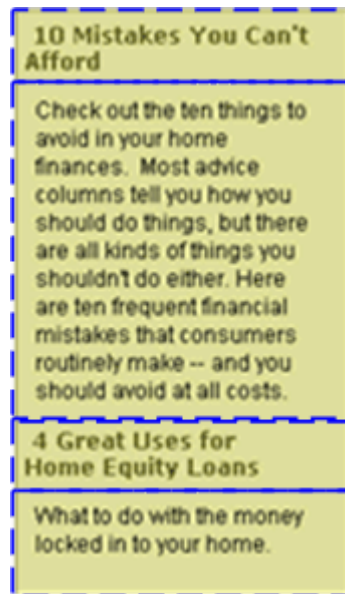
Figure 19: Snippet Template

Noah notes that the output that a Template defines does not have to be a Web page or part of a Web page; it can be information stored in a database or other storage system.

He further explains that Templates designed for use on Web Sites are either type-specific (defined for a single Content Type) or shared (can be used by multiple Content Types). They have discussed Global Templates, but will not attempt to define the Site's Global template right now. At this point, Ally will begin planning her Page Templates, the Snippet Templates that will be included on her Page Templates, and the Content Type or Content Types each Template is associated with. She must also design her Slots and the Templates that each Slot can contain.

Noah briefly explains that to create a typical Page or Snippet Template that displays a text-based Content Item, an implementer uses an existing HTML page in the Web Site or defines a sample page (or portion of a page) in HTML, adds code for Slots, and then adds the HTML to a new Template using the CM System Workbench's Template wizard and editor. The Wizard uses a templating language (by default, Apache's Velocity) to create the Template, which is then applied to Content Items to create Pages and Snippets in the format of the original HTML that the implementer defined. The Slots referenced in the Template are defined in the Rhythmyx Workbench's Slot Wizard and Editor.

Ally already has HTML code defining the pages that she has reviewed with Noah. Of course, sections that will be coded as Slots in CM System do not yet specify Slots, and she will have to recode them. Noah mentions that they will move any formatting information that is duplicated in the HTML for different pages into a cascading stylesheet. This includes colors, font types and sizes, and margin sizes. This introduces another topic, that Noah briefly mentions because it affects Ally's decisions on how to format the Content Type fields that appear in her Templates. He explains that the fields in CM System Content Types can be local to a Content Type, or shared or system fields that appear in any number of Content Types. It is common to create a default format for shared or system fields in a cascading style sheet, so that they appear in the same format regardless of the Template that displays them. The default format in the cascading style sheet can always be overridden in a Template if she wants the field to appear differently.

By decomposing the pages with pen and paper, they can determine which portions represent fields and which portions represent Slots so that they can modify Ally's existing HTML code. They can also see what is repeated among pages, and change repeated elements into Slots, Templates, and cascading style sheet markup.

Decomposing Four Similar Pages

Questions:
- Specifically, which fields from the page's Content Type do you want to display on each page?
- Which portions of each page do you want to use to display information from other Content Types?
- How much flexibility do you want to give Web Site designers in determining what is displayed on each page?
- How do you want to format the information on each page?

As Ally and Noah have analyzed the pages on Ally's Web Site, they have already determined that the About Enterprise Investments, the EI Global Services Funds, the Funds, and the Products and Services pages should represent a Generic Content Type, but may be associated with different Templates. Noah suggests that they begin by fully decomposing these four similar pages to help them determine the remaining issues.

The following two graphics show how they decompose the About Enterprise Investments and EI Global Financial Service Fund pages on paper. Because they have already analyzed the pages, they are able to easily identify what functions best as a Content Type field and what functions best as a Slot. As they have already discovered, these two pages are structurally the same. They both include the title and body fields from the Content Type, and sidebar and bottom Slots for including information and/or links to other Content Items. Ally is comfortable with the idea of using the structure of the pages as a Generic Template that will be assigned to a Generic Content Type, and possibly other Content Types. The Generic Content Type will have a Title field and a Body field as well as other fields that they will determine in the next step of the modeling and design process.

Figure 20: Decomposed About Enterprise Investments page

Sidebar Slot

Title field

EI GLOBAL FINANCIAL SERVICE FUND

Investment style: A disciplined, bottom-up value approach concentrates a small portfolio of 40 to 50 of the best stock ideas with an expected turnover of about 25%. Extensive interviews are conducted with management in the primary subsectors of banking, insurance, asset management and consumer and commercial finance companies.

Fund Type	Specialty
Management Style	Bottom-up value
Date of Inception	April 2000
Total Net Assets	$16.8 million
Eligibility	RESP, foreign content portion of RSP & RIF

Body field

EI Global Real Estate Equity

The fund's objective is to provide long-term capital growth. It invests primarily in securities within companies, Real Estate Investment Trusts (REITs) and other trusts and entities around the world whose primary business is directly or indirectly related to the ownership, management or development of real estate property. It will not invest directly in real estate.

Investor Suitability: this fund is suitable for investors who wish to diversify their portfolio into specific sectors of the world economy. The fund would complement a large-cap value style fund or a more broadly diversified global equity fund. Check it out at www.investersplace.com .

more >>

Bottom Slot

Related...
EI Resources Fund

The fund's objective is to provide long-term capital growth. It invests primarily in shares in companies engaged in the discovery, development and extraction of oil, gas and metals.

Investor Suitability: Suitable for investors seeking maximum long-term capital growth and who are tolerant of short-term volatility. The fund provides potential to benefit from rising prices of natural resources. The fund is a good complement to a more broadly diversified domestic or international equity fund for added diversification.

Figure 21: Decomposed EI Global Financial Service Fund page

Now Ally and Noah decompose the Funds page. Ally's original decomposition of the Funds page follows. She makes an effort to duplicate fields and Slots in the Generic Template to see if she can use the same Template. She also fills in a place for a Sidebar Slot, which could be added if she decides to use the same Template.
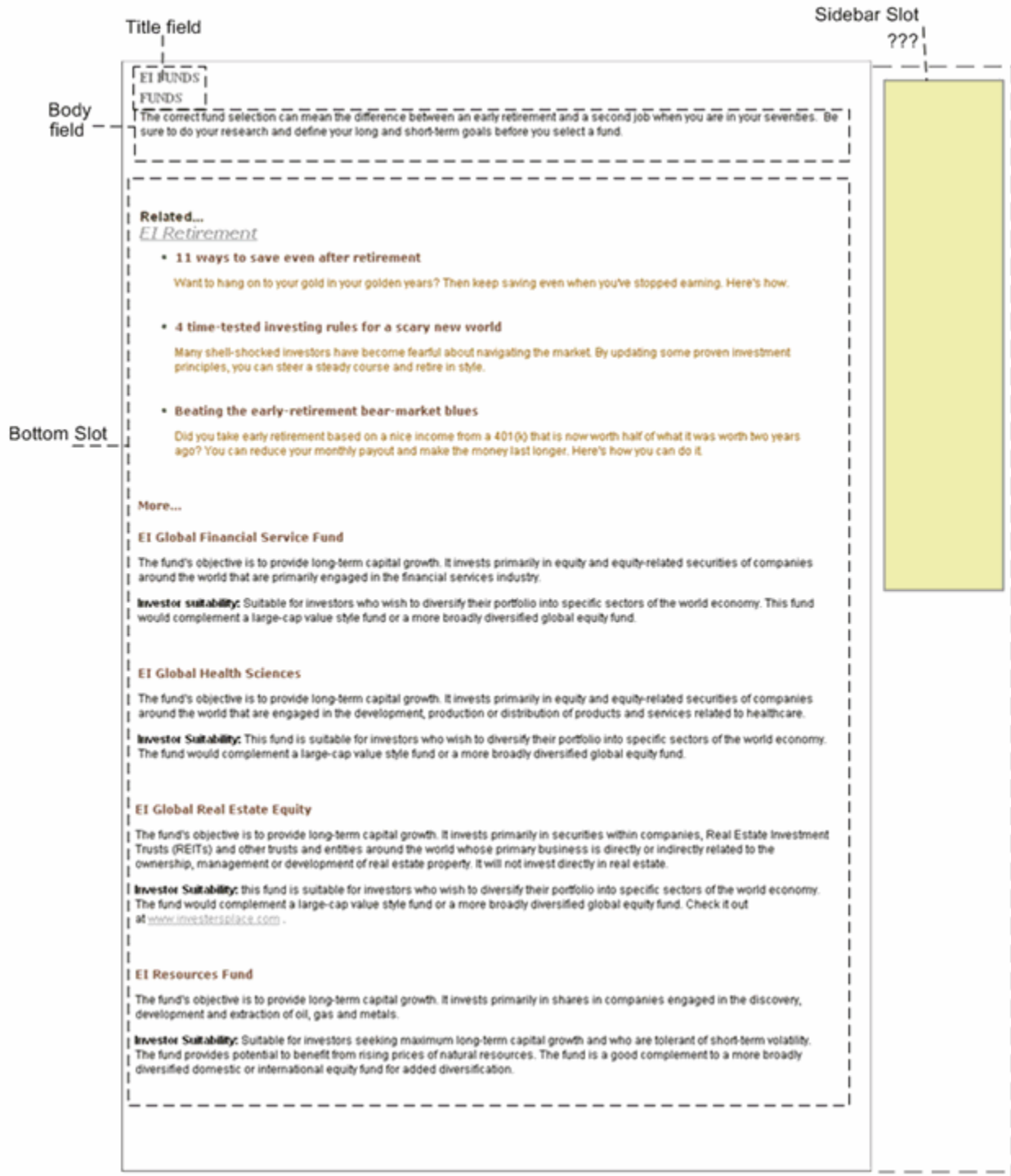
Figure 22: Decomposed Funds Page

At this point, Ally is in favor of making a single Template that performs the rendering and the formatting for the three pages that they have decomposed so far. She believe this will make their system simpler yet offer her company's Web Site designers a fair degree of flexibility.

Noah agrees that this is a good solution. Now he suggests that they decompose the Products and Services page since they have already determined that due to the columns in the bottom Slot, a separate Template may be required.

Ally includes the same Title field and Body field that she has included in her other Templates. She includes the possible Sidebar Slot at the right side of the page, and at the bottom of the page, she makes a note that the Bottom Slot will have a table option.
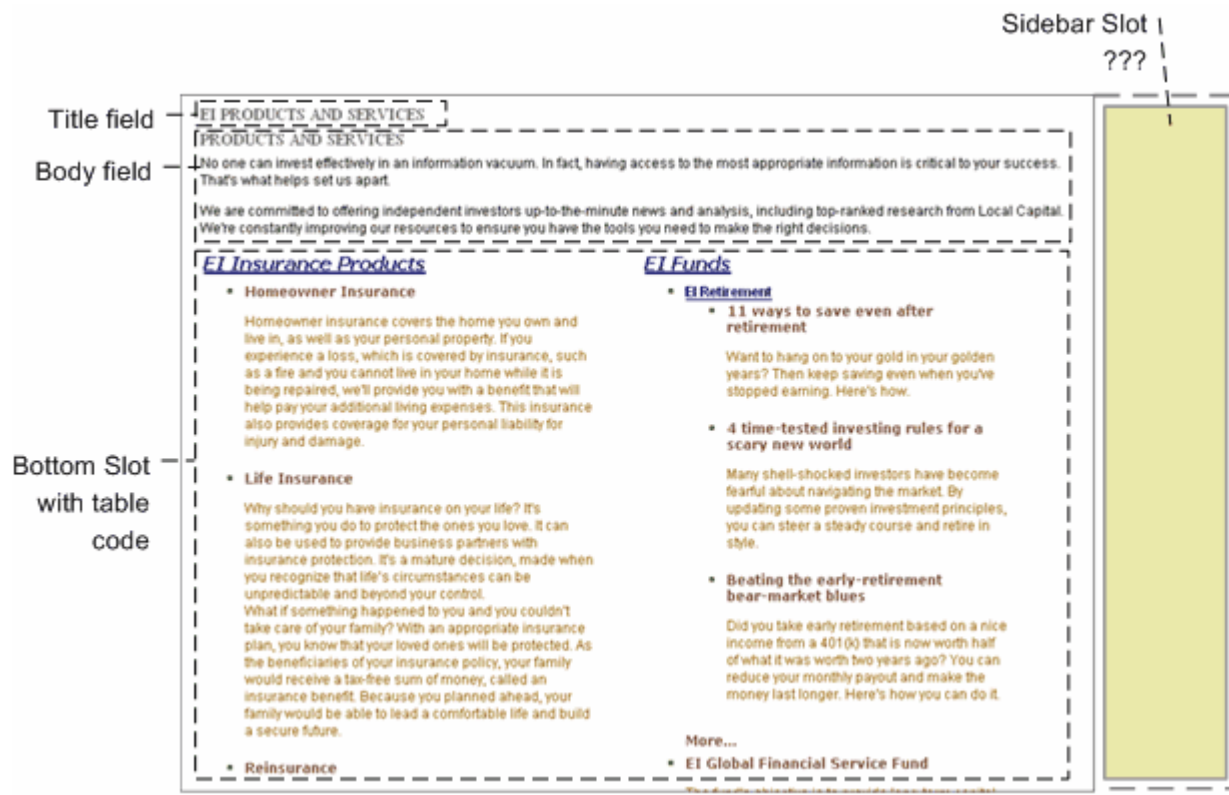
Figure 23: Decomposed Products and Services page

Noah now explains to Ally why the double column approach as used here will require a different Template. In the other content items, each item inserted into the Bottom Slot is displayed vertically. For this content item, each item inserted into the Bottom Slot is displayed horizontally across the page as the top item in a column. In the example above, these items are *EI Insurance Products* and *EI Funds*. They are displayed at the top of each column as a title link. The items listed below the top item in each column are included in the top item's Bottom Slot (in the example above, the *EI Insurance Products* item includes the *Homeowner Insurance*, *Life Insurance* and *Reinsurance* items in its Bottom Slot. The *EI Funds* item includes the items below it in its Bottom Slot). Although both Templates can use the same Bottom Slot, some code to specify the table format must be included in one of the Templates to cause the Slot to display the items horizontally.

Ally has one concern - how will CM System know which page Template to use for each Generic content item? Noah says that CM System has a special type of Template called a Dispatch Template created just for this purpose. A Dispatch Template does not include any code or markup. It simply looks at a condition, applies it to the appropriate content item, and depending on the result of the condition, chooses the correct Template. For example, Ally could add a *Usage* field to the Generic Content Type that specifies whether or not the item defines a major product. Depending on the value of this field, CM System would create a page using the normal Generic Page Template or the Generic Page Template with the horizontal list at the bottom.

Noah has a few minor suggestions about component names. He suggests that she change the name "Bottom Slot" to "List Slot" in order to give the page designer an idea of its intended function. In most pages where this is used, the Slot contains a list of at least a few items that the user can go to for more information. In addition, he suggests that she change the name of the title field to displaytitle, since they will probably include a separate system title for each Content Type, and displaytitle clarifies the function of this title.

Now Ally has some Template designs that she can use to begin to implement her system. Of course, these templates may change as she does more planning and begins implementing. Her current normal Generic Template is the following:
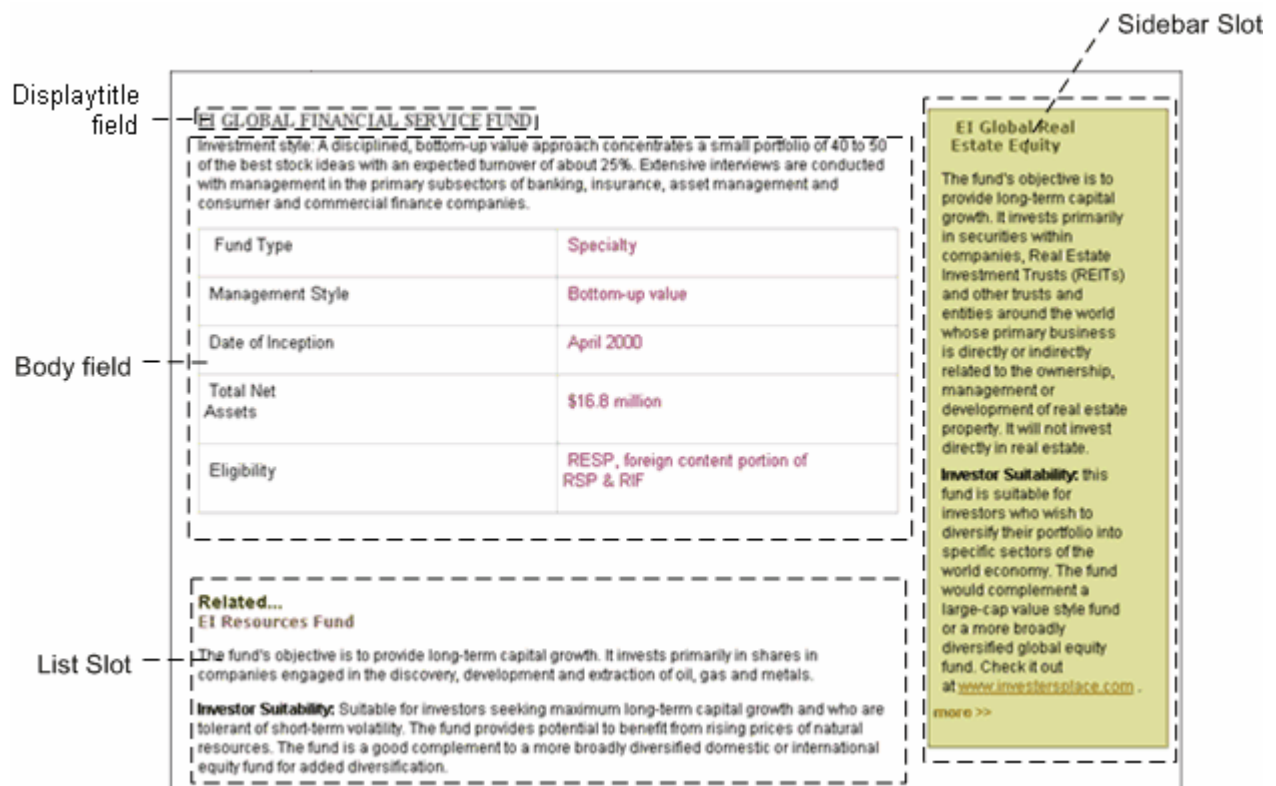
Figure 24: Generic Content Type Template

Her current Generic Template for major products is the following:
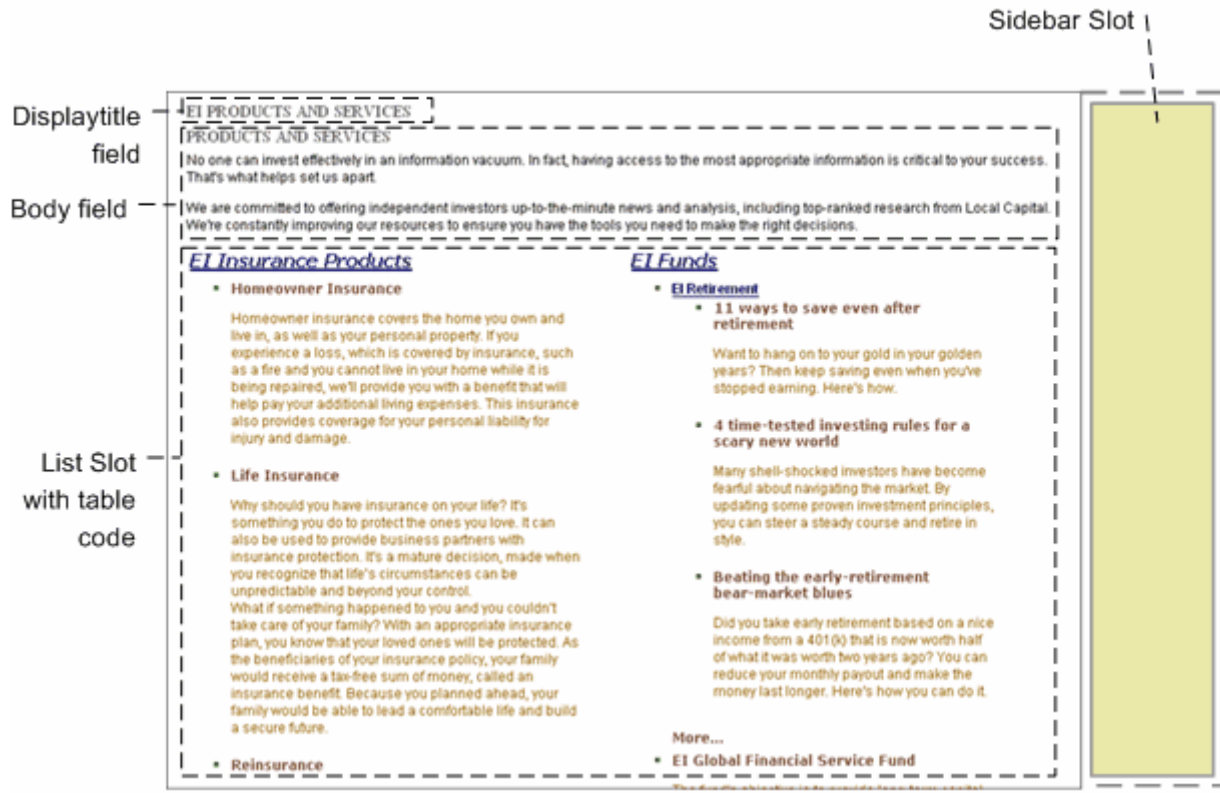
Figure 25: Generic Template for Major Products

Decomposing the Events Page

Ally and Noah have already determined that the Events Page will represent an Event Content Type and an Event Page Template. Previously, she told Noah that she wanted to change the page to show all of the locations (and the details of each location) where the event will take place. However, she does not know how a CM System field is designed to hold multiple locations and their details.

Noah explains that multiple locations and their details are not stored as a Location field but as a Location child field set in a CM System Content Type. A child field set allows for multiple entries in a table format, and the field set supports any number of fields. Ally's Location child field set would have a format similar to:

	City	**State**	**Address**	**Contact**
Location 1	City 1	State 1	Address 1	Contact 1
Location 2	City 2	State 2	Address 2	Contact 2
Location 3	City 3	State 3	Address 3	Contact 3
. . .	. . .	. . .	. . .	. . .

The procedure for adding this child field set is shown in the *Creating Shared Fields* chapter of the *CM System Implementation Guid*e. Of course the format of Ally's original Page must be changed to list all locations in the child field set as shown in the following graphic. The procedure for creating the Template that displays all of the contents of a child field set is shown in the *Creating Slots and Templates* chapter of the *CM System Implementation Guide*.

They decompose the Events page to determine the structure of the Template and to begin to determine which fields are required on the Content Type:
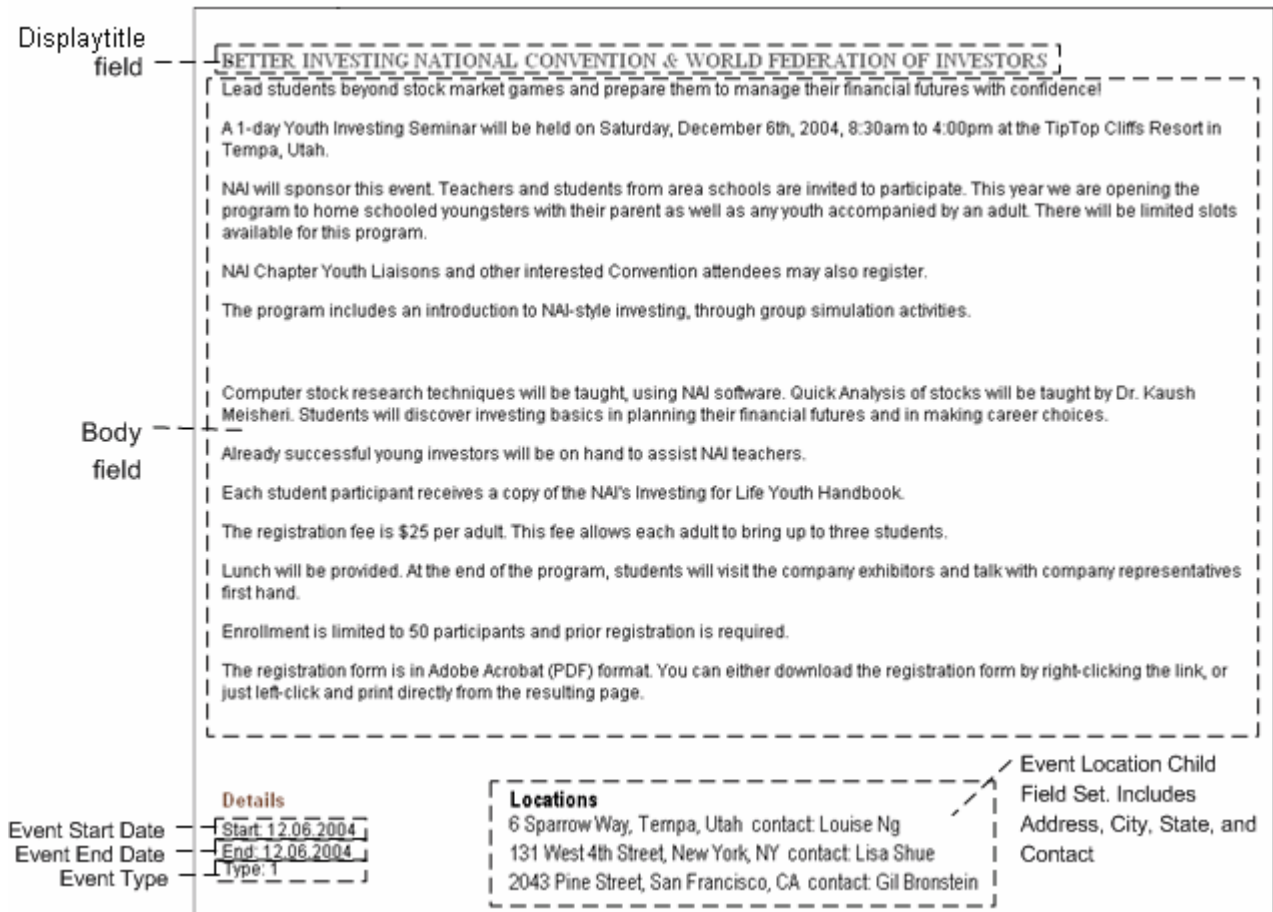
Figure 26: Decomposed Event Page

Noah believes this will translate easily into a Template. His only suggestion is that Ally may want to include a List Slot at the bottom of the page which could optionally display links to pages of related information. Ally agrees with this suggestion, and the final Template appears as:

Displaytitle
field

BETTER INVESTING NATIONAL CONVENTION & WORLD FEDERATION OF INVESTORS

Lead students beyond stock market games and prepare them to manage their financial futures with confidence!

A 1-day Youth Investing Seminar will be held on Saturday, December 6th, 2004, 8:30am to 4:00pm at the TipTop Cliffs Resort in Tempa, Utah.

NAI will sponsor this event. Teachers and students from area schools are invited to participate. This year we are opening the program to home schooled youngsters with their parent as well as any youth accompanied by an adult. There will be limited slots available for this program.

NAI Chapter Youth Liaisons and other interested Convention attendees may also register.

The program includes an introduction to NAI-style investing, through group simulation activities.

Body field

Computer stock research techniques will be taught, using NAI software. Quick Analysis of stocks will be taught by Dr. Kaush Meisheri. Students will discover investing basics in planning their financial futures and in making career choices.

Already successful young investors will be on hand to assist NAI teachers.

Each student participant receives a copy of the NAI's Investing for Life Youth Handbook.

The registration fee is $25 per adult. This fee allows each adult to bring up to three students.

Lunch will be provided. At the end of the program, students will visit the company exhibitors and talk with company representatives first hand.

Enrollment is limited to 50 participants and prior registration is required.

The registration form is in Adobe Acrobat (PDF) format. You can either download the registration form by right-clicking the link, or just left-click and print directly from the resulting page.

Event Location Child Field
Set. Includes Address,
City, State, and Contact

Details

Event Start Date — Start: 12.06.2004
Event End Date — End: 12.06.2004
Event Type — Type: 1

Locations
6 Sparrow Way, Tempa, Utah contact: Louise Ng
131 West 4th Street, New York, NY contact: Lisa Shue
2043 Pine Street, San Francisco, CA contact: Gil Bronstein

List Slot

Figure 27: Events Content Type Template

At this point, Ally has determined that the Event Page Template will be type-specific for the Event Content Type, which will include the fields Displaytitle, Body, Event Start Date, Event End Date, Event Type, as well as others that they will determine in the next modeling and design step, and the Event Location child field set.

C H A P T E R   5

Refining Content Types and Shared Fields

> **Questions:**
> - Which information on the pages and snippets that you have sketched out represents Content Type fields?
> - What system information do you want to store in each Content Type?
> - Which fields are repeated in multiple Content Types?

Now that Ally has determined her system's Content Types and the fields that will appear on the Templates for the Content Types, she is ready to define the Content Types in detail. Noah explains that detailed definitions are necessary in order to create a development plan that another user can implement. He begins by reiterating the components that make up Content Types and the properties associated with them.

Noah explains that each Content Type is composed of fields, which CM System categorizes as local, shared, or system. During creation of a Content Type in the Workbench, implementers can define local fields that are specific to a Content Type. Implementers can also add user-defined shared fields and CM System-defined system fields to any Content Type. Implementers usually define non-system fields that are used by most Content Types in their system as shared fields. This saves time when defining Content Types and ensures that common fields are stored in consistent formats. After Ally defines the fields in her Content Types, she will be able to see which of them are used frequently enough to include in shared field sets (shared fields that are stored in a set because they are related).
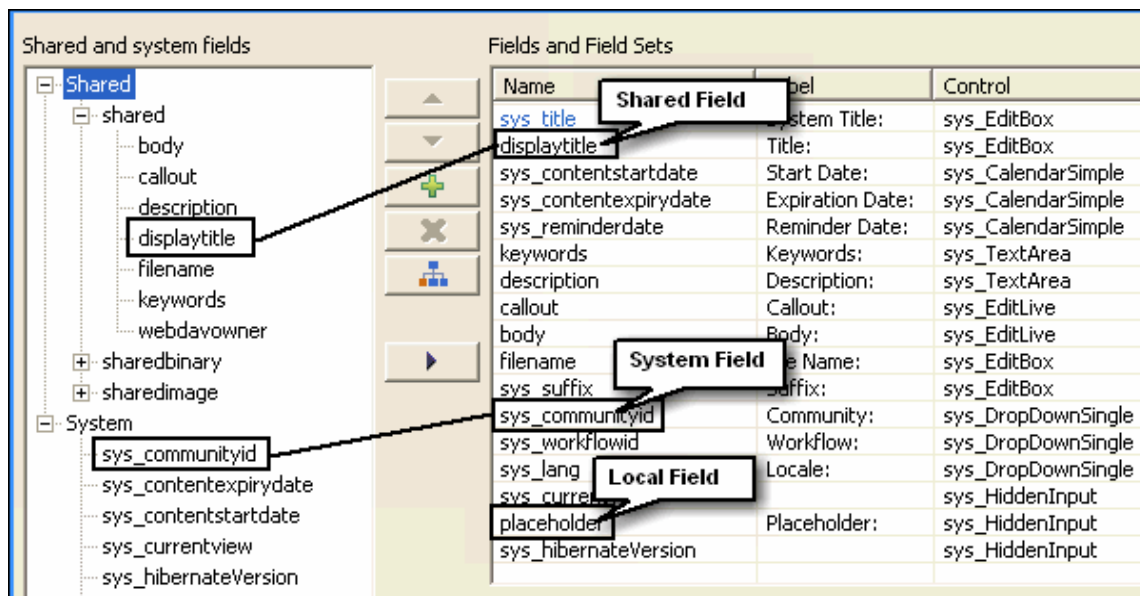
Figure 28: Content Types can include shared, system and local fields

One or more Workflows are permitted for each Content Type, and a default Workflow is required. Otherwise, Content Items of that type could never be approved and entered into a publishable State.

Each Content Type must also be visible to one or more Communities to enable users to have access to it.

A Content Type can exist without a Template associated with it, but without a Template its Content Items can never be published since Content Items are always published in the format of a Template.

In summary, Noah states that for each of her Content Types, Ally must specify:

- shared fields
- system fields
- local fields
- field validations and visibility
- allowed Workflows
- the default Workflow
- Communities that have access to it
- Templates

Of course, during the implementation process, she may change some of these specifications.

Ally has already *designed the Templates* (see page 35) associated with each of her Content Types. She will determine the Communities and Workflows for her Content Types in the section *Planning Roles, Communities, and Workflows* (see page 95). In this section, she will focus on determining the fields and shared field sets in her Content Types and defining their properties.

Determining Fields for the Generic Content Type

Ally and Noah begin by deciding what fields to include in the Generic Content Type. Initially they look at the page Template that they analyzed for the Generic Content Type. Early in the **Web Page decomposition process** (see page 27) they determined that the Generic Content Type should include the following fields:

- displaytitle
- body
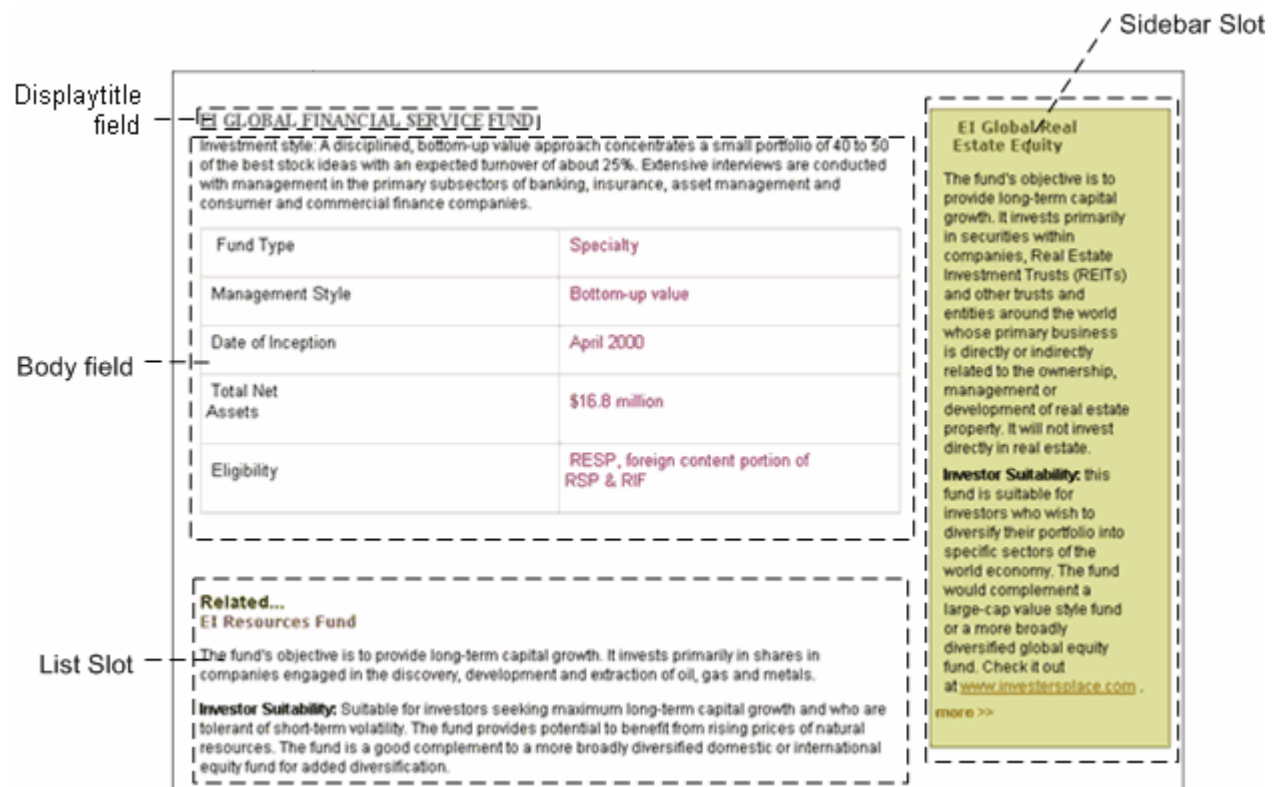
Figure 29: Generic Content Type Template

In addition, they have designed some Slots on other pages that will hold information from Generic Content Items. In one of these Slots (shown in the green sidebar, above), they have seen a title and marketing blurb (callout) for a Generic Content Item.

They add the callout field to the Generic Content Type:

- callout

Noah tells Ally that certain system fields are required in all Content Types, and in fact, the Workbench automatically adds them to new Content Types when they are created. They add these system fields to the list of fields for the Generic Content Type:

- sys_title (the internal title for the Content Item)
- sys_communityid (the login Community of the user who created the Content Item, that is, the Content Item's Community)
- sys_lang (the login Locale of the user who created the Content Item, that is, the Content Item's Locale)
- sys_currentview - (a field that CM System uses internally)
- sys_workflowid - (the Workflow associated with the Content Item)
- sys_hibernateVersion - (a field that CM System uses internally)

Noah mentions that implementers frequently add the system fields sys_contentstartdate, sys_contentexpirydate, and sys_reminderdate to all Content Types. These fields designate when the Content Item should become public, when it should be archived, and a date for sending a reminder to certain users (for example, a date past sys_contentstartdate to send to users who have not approved the item). He also says that implementers often also include the sys_suffix field, which is used to generate the location where Content Items are published. Ally agrees to add these four fields to the Generic Content Type as well as her other Content Types:

- sys_contentstartdate
- sys_contentexpirydate
- sys_reminderdate
- sys_suffix

Why include sys_suffix:

The CM System publisher uses a formula referred to as a Location Scheme for determining the path and filename to use for each Content Item it publishes. A Location Scheme can be a literal file name, such as `index.html`, or a formula, such as the following expression:

```
'item' +
$sys.item.getProperty('rx:sys_contentid').String + '.'+
$sys.item.getProperty('rx:sys_suffix')
```

This expression could translate to item456.htm or item333.jpg. The final element in the expression, $sys.item.getProperty('rx:sys_suffix') tells the Publisher to use sys_suffix as the suffix or extension for the filename, and this is why sys_suffix is included in Content Types.

Note: Java Expression Language (JEXL). JEXL expressions for Location Schemes are often more complicated than the examples shown. For details about JEXL, see http://jakarta.apache.org/commons/jexl/. (see akarta.apache.org/commons/jexl/ - http://)

Noah points out that in some cases, the Content Item may already have a file name. For example, if it is a binary Content Type uploaded through a sys_File control, or a Content Item created in another application and uploaded through WebDAV. In these cases, she may want to store the file name, either to keep a record of the original uploaded file name, or to use it in the Location Scheme.

For example, the Location Scheme could be:

```
'item' + $sys.item.getProperty('rx:filename').String
```

This scheme would show which Content Item was published by indicating its uploaded file name rather than its Content ID.

Ally decides to add a filename field to her Content Types for potentially storing each Content Item's publishing filename.

- filename

Ally explains that she wants potential visitors to be able to find her pages in search engines and wants fields for storing search keywords and a description of the page that the search engine can use. She adds keywords and description fields that can hold content that will be added to the page source code for this purpose.

- search keywords
- search description

Noah says that Ally is required to add at least one local field to each Content Type. Earlier, they discussed adding a field that a *Dispatch Template* (see page 38) could use to determine the function of a content item. They add a *usage* field, which will hold a value to tell the Dispatch Template whether or not the content item stores a main product category.

- usage

The bulleted names in this topic represent the fields in Ally's Generic Content Type. Later, she will assign properties, such as data type and size to these fields, and eventually, she will add them to the development plan for her CM System CMS. You can see these fields listed together with descriptions of their properties in the *development plan* (see page 117).

Now she will complete a similar process to define the fields in her Event Content Type.

Determining Fields for the Event Content Type

When Ally defined the fields that would comprise her Generic Content Type, she determined that many of these fields would be included in all of her Content Types. So she begins selecting the fields for her Event Content Type by listing the fields that she plans to use globally. For explanations of these fields, see ***Determining Fields for the Generic Content Type*** (see page 49):

- sys_title
- sys_communityid
- sys_lang
- sys_currentview
- sys_workflowid
- sys_hibernateVersion
- sys_contentstartdate
- sys_contentexpirydate
- sys_reminderdate
- sys_suffix
- filename

Now they look at the page template that they have designed for the Event Content Type.
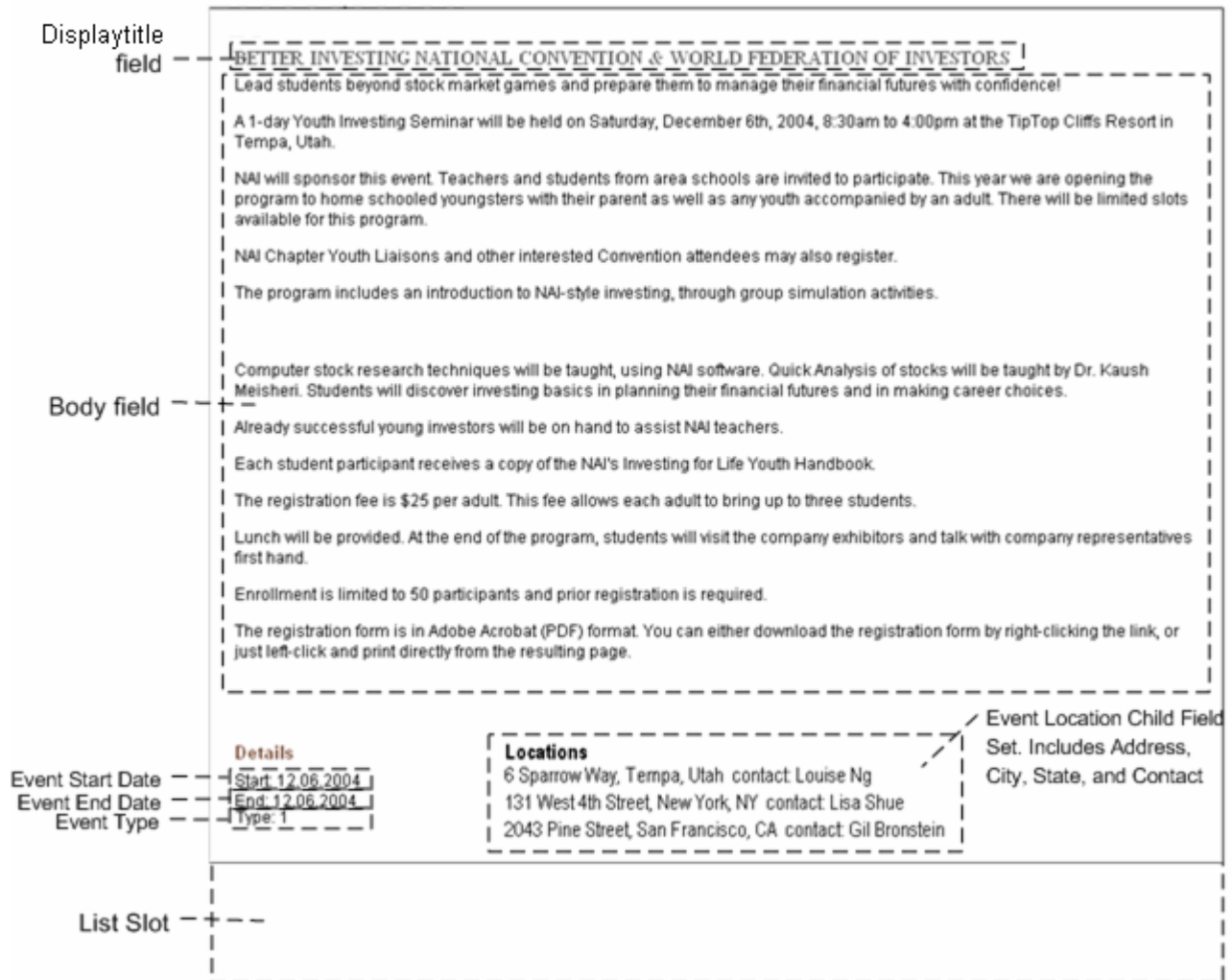
Figure 30: Events Content Type Template

This page includes the following fields:

- displaytitle
- body
- event start date
- event end date
- event type
- multiple locations that show:
    - event address
    - event city

- event state
- event contact

In the topic ***Decomposing the Events Page*** (see page 44), Noah has already introduced the idea of the multiple locations being stored as a child field set (a table of values) and address, city, state, and contact becoming fields in each column in the location child field set.

The same Slot that they have designed to hold a title and marketing blurb (callout) of the Generic Content Type can also hold a title and callout of the Event Content Type, so Ally adds a callout field.

- callout

In addition, Ally wants potential site visitors o also be able to find her Event pages in search engines, and adds the same keywords and description fields that she included on her ***Generic Content Type*** (see page 49):

- search keywords
- search description

The fields for the Event Content Type appear to be planned, but Ally explains that at some point, Event Content Types may be composed in an application outside of CM System, and she may wish to use the open source WebDAV feature to upload content from outside applications into Event Content Types. She asks if this requires any additional fields. Noah says that to use WebDAV, she is required to include a webdavowner field to hold the name of the user who currently is using WebDAV to upload the file. So she adds the field to her Content Type:

- webdavowner

The bulleted names in this topic represent the fields in Ally's Event Content Type. Later, she will assign properties to these fields, and eventually, she will add them to the development plan for her CM System CMS.

Now she will complete a similar process to define the fields in her Image Content Type.

Determining Fields for the Image Content Type

Ally begins defining the fields for her Image Content Type by listing the fields she plans to use globally.

For explanations of these fields, see ***Determining Fields for the Generic Content Type*** (see page 49):

- sys_title
- sys_communityid
- sys_lang
- sys_currentview
- sys_workflowid
- sys_hibernateVersion
- sys_contentstartdate
- sys_contentexpirydate
- sys_reminderdate
- sys_suffix
- filename (Notice below that the *img1_filename* field automatically stores the extracted filename; however, a standard location scheme could use the field *filename*, and then it is important that the *img1_filename* value is moved into the *filename* field. Currently no location schemes use the *filename* field.)

Earlier, Noah mentioned that a binary file and its mime type must be retrieved and stored to make an image available to Templates. He also mentioned that the extension (java plugin) that processes the uploaded image extracts additional information that she can store in the Image Type. Now he goes into more detail about the information the extension extracts.

Noah explains that implementers use the sys_fileInfo and/or the sys_imageInfoExtractor extensions to process image files and determine some of their metadata. The only difference between these extensions is that sys_imageInfoExtractor extracts height and width information in addition to all of the other information that sys_fileInfo extracts. When an implementer adds a field that uploads an image to a Content Type, CM System automatically add sys_FileInfo as a dependency of the Content Type. Therefore, to store height and width information, the implementer must add sys_imageInfoExtractor as a preprocessing extension.

The control used to upload the file determines whether or not the field that uploads the file must have a predetermined name. Noah explains that CM System provides two controls for uploading files: sys_File is simple, enabling the user to locate and upload a file, and sys_WebImageFX is more complex, enabling the user to upload, create, or edit a file. sys_webImageFX requires that the field that uploads the image be given a specific filename; sys_file allows the implementer to give the field any name. In either case, the implementer must give the fields that store the metadata extracted the name of the upload file field as a prefix. For example, if the field that stores the uploaded file is named *img1*, the field that stores the filename must be *img1_filename*. The metadata that sys_fileInfo and sys_imageInfoExtractor extract is represented by the following suffixes, which the implementer appends to the upload file field name. Beside each suffix is a description of the content it stores, and an explanation of whether or not the metadata is required for storing or displaying the image. For more information about the sys_fileInfo or sys_imageInfoExtractor extensions, see the *CM System Technical Reference*.

Suffix	Content	Sample Field	Sample Content
_filename	The original filename of the uploaded file. Required by image editing control to display the image. Also often used in system processing.	img1_filename	IMG_0357.JPG
_ext	The file extension. Required by the simpler image upload control to display the image in a browser.	img1_ext	.JPG
_type	The MIME type and subtype. This is required for displaying the image in a browser.	img1_type	image/pjpeg
_size	The length of the file, in bytes. Not required to display the image, but often used in system processing.	img1_size	548650
_height	*Used by sys_imageInfoExtractor only.* The height of the image. Not required to display the image, but often used in system processing.	img1_height	1600
_width	*Used by sys_imageInfoExtractor only.* The width of the image. Not required to display the image, but often used in system processing.	img1_width	1200

Ally does not see any reason to use the more complex sys_webImageFX control, since her users will only be uploading files and not editing or creating them, so she will use the sys_file control and give the field that uploads the file a name that she finds appropriate.

Of course, Ally does not have to use all of the fields that sys_fileInfo and sys_imageInfoExtractor extract in the Image Content Type. Noah tells Ally to add only those that will serve a purpose to her Image Content Type. Ally decides to add the following fields. She uses img1 as the prefix because she also wants to add a thumbnail of the image, for which she will use the prefix img2.

- img1
- img1_filename
- img1_ext
- img1_type
- img1_size
- img1_height
- img1_width

She includes img1_ext and img1_type because they are required for displaying the image in a browser. She decides to include img1_filename since she is storing the filename in all of her Content Types. She also decides to include img1_size, img1_height, and img1_width, which are not required, to help Web designers determine how to best place these images on a page.

Since she wants the option of adding a thumbnail of the image when she uploads it, she also decides to add a second set of the same type of fields:

- img2
- img2_filename
- img2_ext
- img2_type
- img2_size
- img2_height
- img2_width

Ally wants to encourage content contributors to provide alternate text for images that are not rendered and for users who use screen reading software. She adds the following field to hold alternate text for images:

- img_alt

In addition she wants to add an image category field, so that Web designers can sort images by categories when searching in CM System for one to use. She adds a field to hold categories:

- img_category

Noah suggests that she add two more fields. Since her other Content Types have an external title field, he suggests she make this a requirement for all of her Content Types. It may be easier to identify Content Items in the CMS interface by an external title. In addition he suggests that she add a description field not for Web searches, but so that internal users have text they can match when searching for an image. Ally adds these two fields:

- displaytitle

- description

Ally guesses that some of her image files may be uploaded through WebDAV, and adds a webdavowner field. See *Determining Fields for the Event Content Type* (on page 52) for an explanation of this field.

- webdavowner

The bulleted names in this topic represent the fields in Ally's Image Content Type. Later, she will assign properties, such as data type and size to these fields, and eventually, she will add them to the *development plan* (see page 117) for her CM System CMS.

Ally has completed determining the fields for her Content Types. She already knows which of these are system fields. Now she can work with Noah to determine which of these are common enough to be included in shared field sets.

Organizing Shared Field Sets

Questions:
- Which fields are repeated in multiple Content Types?
- Which fields in current Content Types may it be useful to store for creating similar Content Types in the future?

Now Ally has a list of the fields that composes each of her Content Types. She already knows which of these fields are system fields. Noah suggests that they begin to determine which of these fields they want to add to shared field sets. Shared fields are stored as sets within a shared field file. A shared field set generally includes related shared fields.

They can begin by seeing which of the fields appear in most of her Content Types. (During an actual modeling and design session, Ally would have more than three Content Types, so fields that appear in "most" Content Types would normally appear in more than two of them.)

Ally lists the following fields which are common to at least two of the Content Types:

- body
- callout
- title
- description
- filename
- keywords
- webdavowner

Ally groups these fields into a field set, and names it the *shared* field set.

Noah suggests that they group most of the remaining Image Content Type fields into a field set. Several of the fields are required, and Ally has mentioned specific reasons for including those fields that are not required. In the future, if she decides to create another image Content Type, it will be easier to have all of the fields that she wants included already grouped together in a field set. Ally agrees, and they group the following fields into another field set:

- img_alt
- img1
- img1_ext
- img1_filename
- img1_height
- img1_size
- img1_type
- img1_width
- img2
- img2_ext

- img2_filename
- img2_height
- img2_size
- img2_type
- img2_width

Ally names the field set *sharedimage*.

The remaining fields that are not system fields are local fields. These are:

- the Generic Content Type's *placeholder* field
- the Image Content Type's *img_category* field
- the Event Content Type's *event_start*, *event_end*, *event_type* fields and the fields in the *event_location* child field set.

Now that Ally and Noah have determined her shared field sets, they proceed to assign properties to all of her shared and local fields.

Assigning Properties to Fields

Questions:
- What basic properties, such as size, data type, and control, best suit each field?
- What special properties, such as visibility conditions and search rules, do you want to apply to certain fields?
- Is a default value necessary?

Now that Ally and Noah have chosen the shared and local fields for the Content Types in Ally's CMS, they must specify the properties of the fields. Noah lists the properties that they should determine for each field:

- Content Editor label that appears beside the field
- Content Editor control for displaying field
- data type
- size
- required/optional
- default value
- whether or not to enable searching on field
- read-only rules
- rules for hiding field
- rules for validating field
- rules for transforming the field's format
- whether or not to allow field to be shown in metadata views of Content Items in Content Explorer
- For field sets:
    - requirements for number of rows entered
    - whether or not to enable users to reorder rows entered
    - whether or not to show field on main Content Editor page

Noah mentions that other field properties exist that it is not necessary to determine now because the value can just as easily or more easily be decided during implementation. These properties include the value of the keystroke for accessing the field (the mnemonic)

Noah suggests that they begin with their field sets, and then go on to the local fields in their Content Types. The properties for system fields are already defined.

Assigning Properties to Shared Fields

Noah and Ally begin by assigning properties to the fields in the shared field set. The shared field set consists of the following fields:

- body
- callout
- displaytitle
- description
- keywords
- filename
- webdavowner

body Field

They begin by looking at the body field. Looking at the way it appears in the page Templates in some of Ally's sample Content Items, they agree that it should be a large text field:
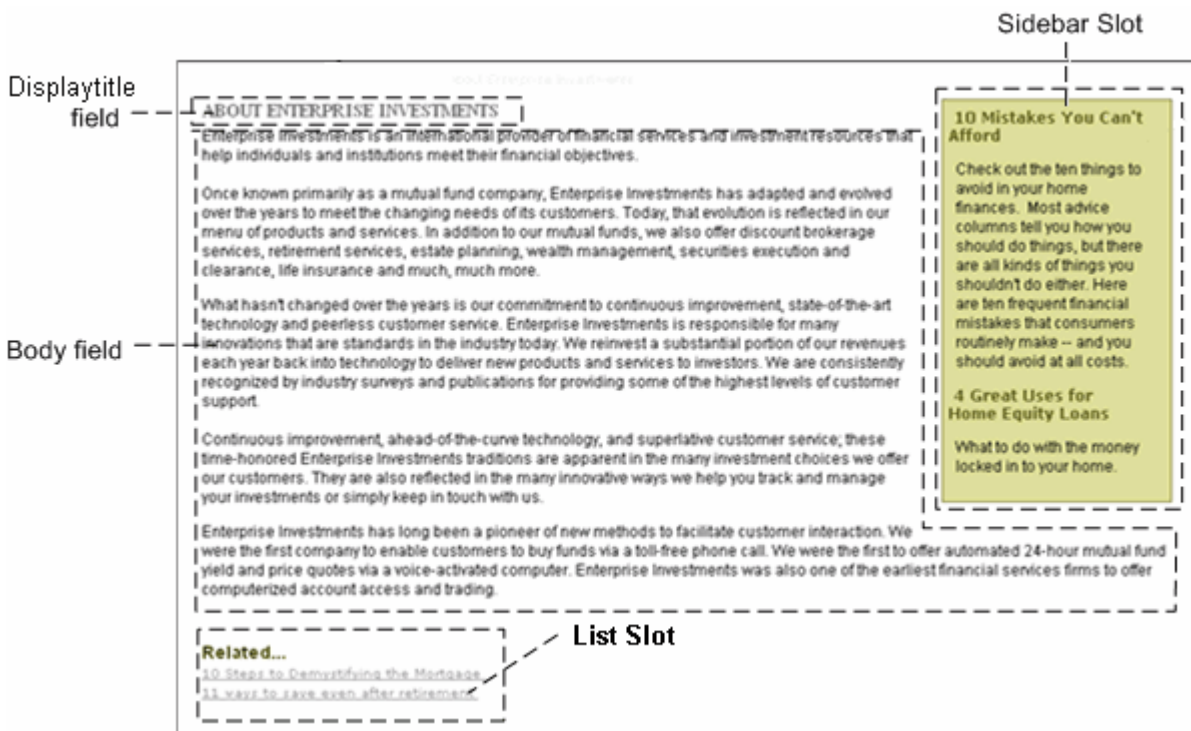
Figure 31: About Enterprise Investments page
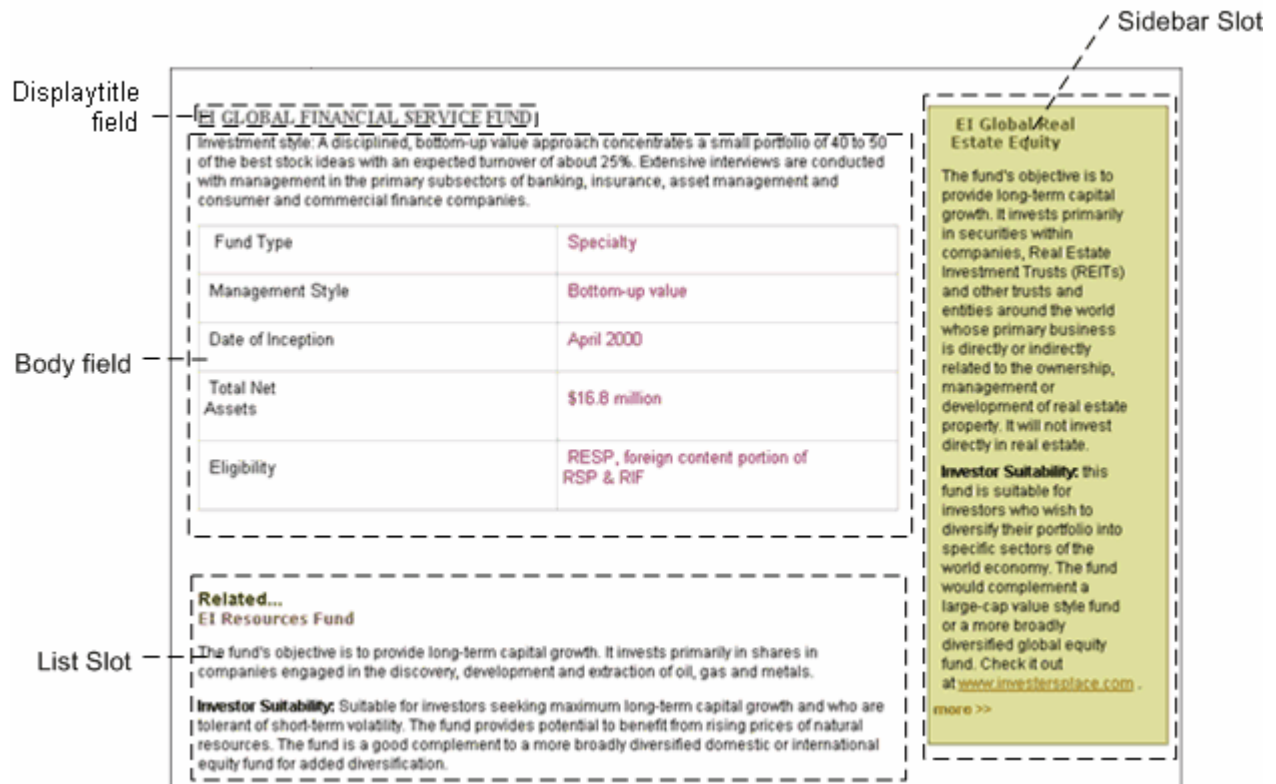
Figure 32: Generic Content Type Template
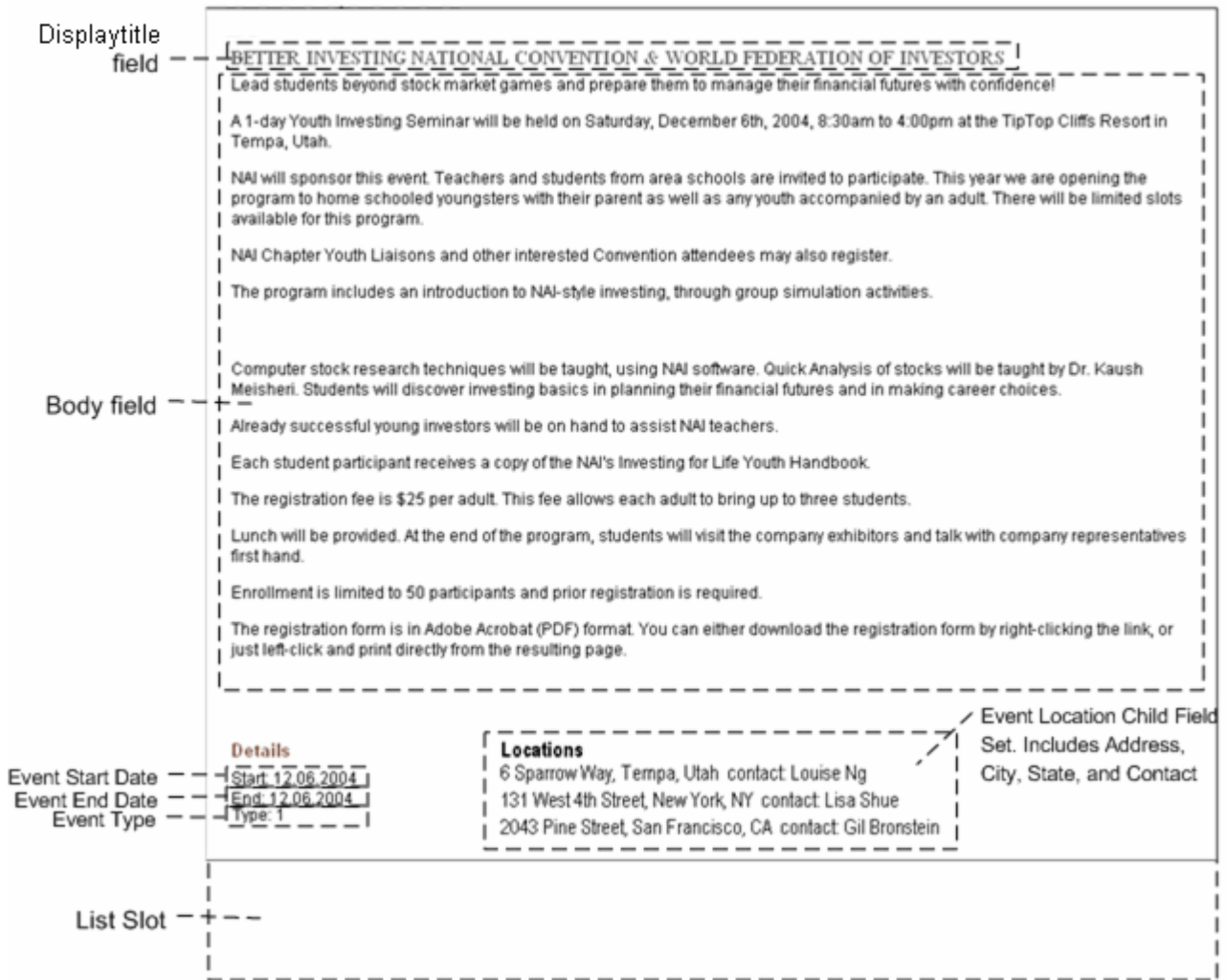
Figure 33: Events Content Type Template

Earlier, when they were ***initially decomposing Web pages*** (see page 27), Ally had mentioned that the EI Global Financial Service Fund page includes a formatted table in the body field. Noah suggested that CM System could handle this if Ally used a rich text control for the field, so content contributors could optionally format text in the body. Noah shows her the sys_EditLive control, CM System's rich text control, which uses the Ephox EditLive for Java editor:
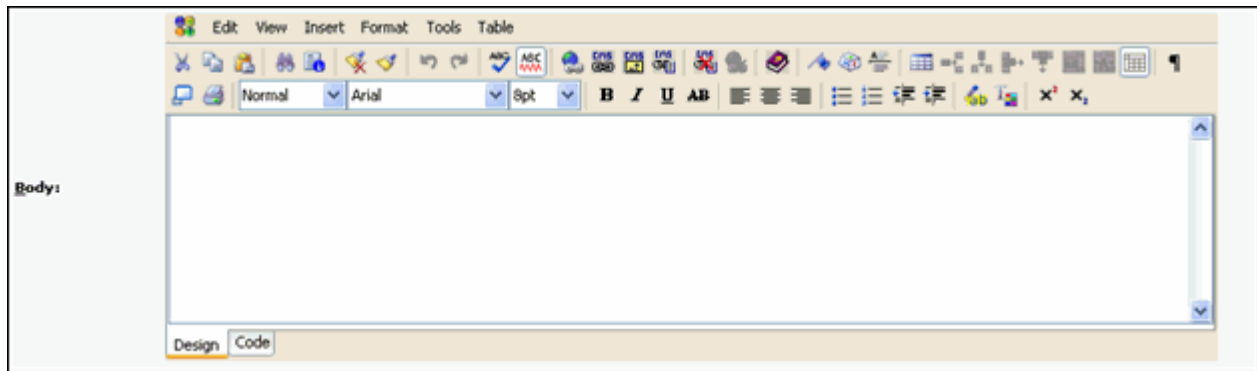
Figure 34: Body field

The sys_EditLive control enables users to enter formatted text (and even graphics) into the field. For fields that use a sys_EditLive control, a data type of *text* and a size of *max* (the maximum the database can hold for the data type) are required.

Noah adds that any number of fields in a Content Type can use the sys_EditLive editor, but when users view the fields in the content editor, only one sys_EditLive field can be active and edited at a time. When a user first opens a content editor including a sys_EditLive field, the user must click on the field to enable it. Once the user clicks on another sys_EditLive field, the first sys_EditLive field becomes disabled.

Ally decides that they can use the field's name, *Body*, as the label that appears in the Content Editor. Noah asks Ally if she wants to require users to enter data in the Body field in order to save the Content Item. She does not think this is a good idea because users may want to begin Content Types by simply entering the title and then finish them at a later time.

Noah says they now have the basic information for the Body field:

Name	Label	Control Name	Occur	Data Type	Format (size)
body	Body:	sys_EditLive	optional	text	max

Now he asks Ally about some specific details that must be entered when creating the field.

- Noah asks her if she wants the field to have a default value. She does not think this is necessary. Noah agrees, and adds that default values are usually used when a field is required, for example, a required start date field may have the current date as its default value.

- He also asks her if she wants user searches to look at the data in this field. Ally definitely wants users to be able to search on this field, since it is one of the main fields in a Content Type.

- He confirms that she does not want to add rules for hiding the field or validating or transforming the data in the field. Since this is one of the main fields that users enter, and users do the formatting themselves, Ally does not want to apply any of these rules to this field.

- Noah also confirms that she wants the field to be visible in metadata views of Content Items. Ally says that she wants the field to be visible in views.

callout field

Now they look at the callout field. In the first two pages shown above, the green sidebar holds displaytitles and callouts from a few different Content Items. Ally notes that the callout's text length varies - in one case it is very brief, but in another, it is as long as a body field. In addition, in the second example, formatting is applied to the text. Ally and Noah agree that the basic information for the callout field can be the same as that for the body field due to its possible lengthiness and its use of formatting.

Name	Label	Control Name	Occur	Data Type	Format (size)
callout	Callout:	sys_EditLive	optional	text	max

Noah asks Ally to decide about the other properties they have discussed: default value, searching on the field, special rules for the field, field visibility in previews.

Ally decides that the values they have determined for these specific details for the body field should be the values they assume for all fields unless they discuss a special reason to treat them differently. Therefore, the callout field will not have a default value, will be searchable, will not include any special rules, and will be visible in previews of Content Items.

displaytitle field

The displaytitle field appears at the top of the pages shown above, and above the content in Sidebar and List Slots. Ally says that she can easily decide that it is a required text field of some brief length, but is not sure which control to use. She wonders if she should use a sys_EditLive control so that content contributors can format it.

Noah says that most implementers choose to maintain the formatting of displaytitle fields in Templates and cascading style sheets for consistency. This leaves Ally with two options for controls, the sys_TextArea control, which allows for several lines of plain text:

Figure 35: sys_TextArea

and the sys_EditBox control, which allows for a single line of plain text:
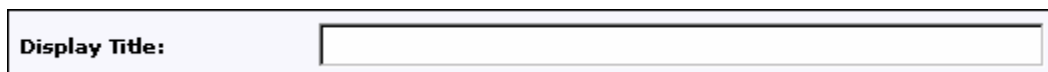
Figure 36: Example sys_EditBox

Since Ally does not want the displaytitle to fill more than one line, she decides to use the sys_EditBox control. Now Ally has the basic information for the displaytitle field, except for the exact size. Noah explains that the size is the amount of space allowed for storing the field in the database; not the size of the control that will appear on the screen. The sizes that controls appear on the screen are set internally in CM System, although the implementer can change the size of controls for individual fields by setting parameters for them in the Workbench. Noah recommends using a generous storage size in case data requires more space to store than expected, so Ally assigns a storage length of 512.

They establish the basic information for the displaytitle field as:

Name	Label	Control Name	Occur	Data Type	Format (size)
displaytitle	Title	sys_EditBox	required	text	512

Ally determines that there are no other special values for other properties assigned to this field.

description field

The purpose of the description field is to provide a description that appears when the page is listed in a Web search engine. It is easy at this point for Ally to decide that it should be an optional text field that uses a sys_TextBox control. Again, Noah suggests specifying a storage size that is much larger than what they will require, such as 1024. No special values for other properties are required for this field. Its basic specifications are:

Name	Label	Control Name	Occur	Data Type	Format (size)
description	Description	sys_TextArea	optional	text	1024

keywords field

The keywords field is similar in purpose to the description field, but provides Web search engines with search words for pages. Ally decides that it can have the same specifications as the description field:

Name	Label	Control Name	Occur	Data Type	Format (size)
keywords	Keywords	sys_TextArea	optional	text	1024

filename field

Ally can easily choose most of the basic information for the filename field. She decides that the field should be an optional text field of 512 characters. She decides by default that she would like to hide the field from content contributors since it is currently not used, although she would like the option to use it in the future. She asks Noah if there is a special control she should use.

Noah shows Ally that there is a sys_HiddenInput control, but it is used to hide fields that should never be visible to content contributors. Instead, he suggests that she apply a visibility rule to the field, since she wants the ability to display the field in some Content Types. The visibility rule used most often to hide a field is a conditional statement that hides the field when 1=2. Of course, this is never true, and the field is not shown. But to make the field visible in certain cases, the implementer can simply change the conditional to 1=1.

Therefore, the filename field has the basic specifications:

Name	Label	Control Name	Occur	Data Type	Format (size)
filename	File Name:	sys_EditBox	optional	text	512

Visibility Rule	
Field	**Rule**
filename	Not visible by default

webdavowner field

Noah notes that when content is uploaded through WebDAV, CM System inserts the name of the user with a lock on the Content Item into the webdavowner field. Therefore, the webdavowner field is similar to the filename field because CM System enters its value rather than content contributors, and in most cases, Ally will want to make it invisible in Content Editors. She gives it nearly the same specifications and visibility rule as the filename field, but changes the size to 255 because a username will probably be shorter than a filename or title field.

Noah suggests that Ally not allow users to search on this field when they perform searches. Since they cannot see the field and it is used internally, if its contents matched a search, it would probably be an accidental match and just cause confusion.

The webdavowner field has the basic specifications:

Name	Label	Control Name	Occur	Data Type	Format (size)
webdavowner	WebDAV Owner:	sys_EditBox	optional	text	255

Visibility Rule	
Field	**Rule**
filename	Not visible by default.

Other properties
Do not enable searching for this field.

Assigning Properties to Sharedimage Fields

Now Noah and Ally assign properties to fields in the sharedimage field set. This field set consists of the following fields:

- img1
- img1_filename
- img1_size
- img1_type
- img1_ext
- img1_height
- img1_width
- img_alt
- img2
- img2_filename
- img2_size
- img2_type
- img2_ext
- img2_height
- img2_width

img1 field

The img1 field is the field that stores the uploaded image file. Ally has two options when choosing the control to upload and store the image file. The sys_File control lets the user enter or search for the file to upload and store. It is compact and easy to use and appears as:

* Image:		Browse...	Preview File	☐ Clear
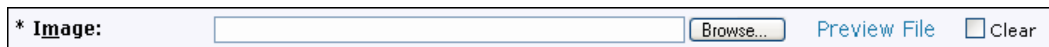
Figure 37: sys_File control

The sys_webImageFX control uses Ektron's WebImageFX editor. It not only allows content contributors to upload graphics, but also to paste them from them clipboard, edit them or design them. Other than the fact that it displays the uploaded file in the WebImageFX editor, the sys_WebImageFX control functions almost identically to the sys_File control. The *sys_imageInfoExtractor and sys_fileInfo extensions return values for the same image properties* (see page 55) and insert them into the same field names; however, the field that uploads the file must be named *uploadfilephoto,* and therefore, all of the meta field names must be prefixed with uploadfilephoto (*uploadfilephoto_type*, *uploadfilephoto_ext*, etc.).

The sys_WebImageFX control is available to users who acquired a special license. For more information about the control, see the *CM System Technical Reference.*
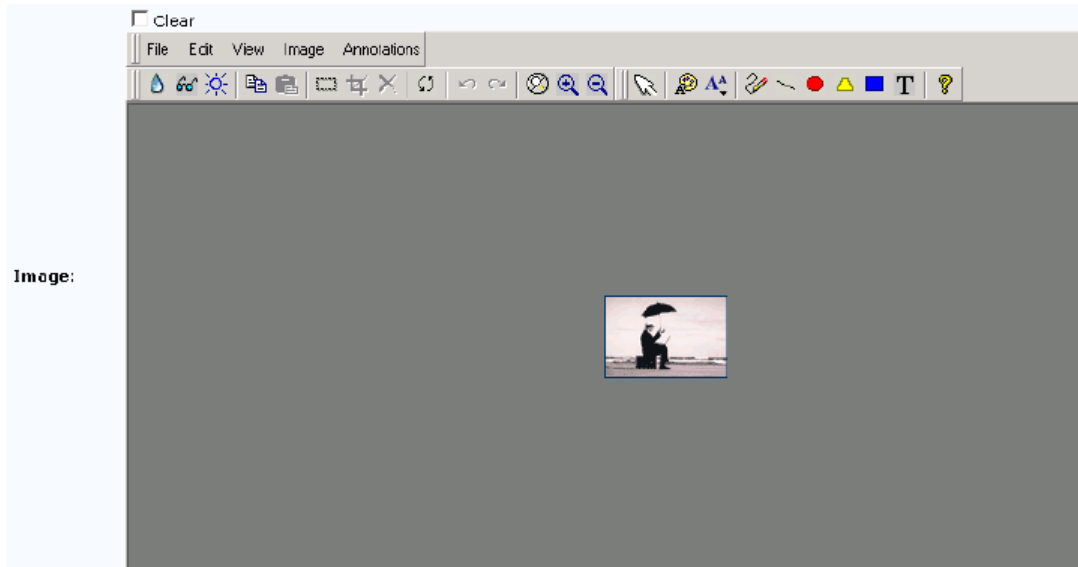
Figure 38: sys_WebImageFX

For both controls, the data type is always binary and the size is max.

Since Ally's content contributors will only select image files to use, but not create or modify them, she has already decided to use the sys_File control. She decides that the field should be required, since the basic function of the Image Content Type is to upload and store an image.

Name	Label	Control Name	Occur	Data Type	Format (size)
img1	Image:	sys_File	required	binary	max

Ally understands that she should disable searching for the field, since this is a binary field, and the search is a text search. In addition, since this is a binary field which may take up a lot of storage, she wants to make sure that she indicates that it should be stored as a binary value in the database.

Other properties
▪ Disable searching.
▪ Store value as binary in database.

img1_filename field

Ally has already created a *filename field for the shared field set* (see page 62), and decides to simply give the img1_filename field the same properties, except this field should be required, since it is filled when the required file is uploaded. In addition, Noah suggests that she not hide this filename, because content contributors may want a reminder of which file they have uploaded.

Name	Label	Control Name	Occur	Data Type	Format (size)
img1_filename	Image File Name:	sys_EditBox	required	text	512

Read Only Rule	
Field	**Rule**
img1_filename	Always read only, since the system enters this field.

img1_size field

This field is relatively easy for Ally to define. Since it holds the size of the file, she decides to make it an integer field that uses the simple sys_EditBox control. Since the size of the file is usually used in internal processing she decides to hide the field, using the 1=2 visibility rule she used with the filename and webdavowner fields in the shared field set. She uses an easily reversible rule instead of a sys_hiddenInput control in case an implementer wants to create a Content Type and make the size visible.

Ally intends to make the img1_size field required, since it will automatically be extracted and inserted with the sys_imageInfoExtractor extension that she will use to upload the file. Noah suggests that she make the field optional, since it is not required to display the image in a browser, and it is possible that in the future a different extension that does not extract the size field will be used to upload images.

Ally agrees to make the field optional, and defines the following properties:

Name	Label	Control Name	Occur	Data Type	Format (size)
img1_size	Image File Size:	sys_EditBox	optional	integer	none

Visibility Rule	
Field	**Rule**
img1_size	Not visible by default

img1_type field

Ally remembers that this field stores the mime type of the uploaded file and is required for displaying the image in a browser. At this point, she has a good understanding of what fields she wants to display, and appropriate data types and sizes for them. She gives the img1_type field the following properties:

Name	Label	Control Name	Occur	Data Type	Format (size)
img1_type	Image Mime Type:	sys_EditBox	required	text	256

Read Only Rule	
Field	**Rule**
img1_type	Always read only, since the system enters this field.

img1_ext field

Like the img1_type field, the img1_ext field is required for displaying the image in a browser. Ally determines to give it the same basic properties as the img1_type field.

Name	**Label**	**Control Name**	**Occur**	**Data Type**	**Format (size)**
img1_ext	Image Extension:	sys_EditBox	required	text	50

Noah suggests that by default she make it invisible with the visibility rule 1=2 since the Content Editor displays the file name, which already includes the extension.

Ally agrees and adds:

Visibility Rule	
Field	**Rule**
img1_ext	Not visible by default

img1_height field

The characteristics and function of the img1_height field is similar to the img1_size field, which Ally has already defined, so she decides to give it the same properties as that field. Noah basically agrees, but has one suggestion: the height (and width) give more significant information to Web page designers than the size because it gives them an idea of how much space the image will take up on a page. Therefore, he suggests that the field not be hidden.

Ally agrees and assigns the properties of the field as follows:

Name	**Label**	**Control Name**	**Occur**	**Data Type**	**Format (size)**
img1_height	Image Height:	sys_EditBox	optional	integer	none

img1_width field

Since the img1_width field has the same function as the img1_height field, Ally gives it the same characteristics:

Name	**Label**	**Control Name**	**Occur**	**Data Type**	**Format (size)**
img1_width	Image Width:	sys_EditBox	optional	integer	none

img_alt field

The img_alt field is a text field that stores a description of an image that a browser uses if the image cannot be rendered or that screen reading software reads. Ally decides to use a sys_EditBox control and store the field as text up to 512 characters in size. She makes the field optional since alternate descriptions for images are not required in her company.

Name	Label	Control Name	Occur	Data Type	Format (size)
img_alt	Image Alt Text:	sys_EditBox	optional	text	512

img2 fields

The fields with prefixes of img2 are used if a web designer requires thumbnails of the img1 images. Ally gives them all of the same properties as the img1 fields, except:

- she changes the word Image in their labels to Mini to indicate that they are thumbnails
- she makes all of them optional since she does not presently plan to use them
- she hides them all with a visibility rule of 1=2.
- she does not make the *img1_type* and *img1_filename* fields read only because they are hidden.

If an implementer decides to use the img2 fields in the future, the implementer must add a special extension (java plugin) to convert the image files to thumbnail size.

We will not list the characteristics of the img2 fields here. If you would like to see their specifications, see the FastForward Implementation Plan.

Assigning Properties to Local Fields

Now that Ally has assigned properties to her shared fields, she must assign them to the local fields in her Generic, Event, and Image Content Types.

Generic Content Type

The Generic Content Type has a single local field named Usage. The function of this field is to tell the **Dispatch Template** (see page 38) the use of the content item so it knows which Template to apply. Ally uses a sys_DropDownSingle control so she can specify two choices for implementers (one choice indicates that the content item is used for a main product category, the other choice indicates that the content item has any other use). She makes it a required text field with a size of 1.

Name	Label	Control Name	Occur	Data Type	Format (size)
usage	Usage:	sys_DropDownSingle	required	text	1

Image Content Type

The Image Content Type has a single local field name img_category. This field stores the category of the image for use in searches or other types of selection. A user enters information into this field by choosing from a set of values. Noah explains that CM System offers several different controls that Ally can use for this type of field.

- ▪ sys_DropDownSingle

   The sys_DropDownSingle control lets users choose one value from a predefined list. The drop list format of this control allows it to be compact, yet include a large number of options.
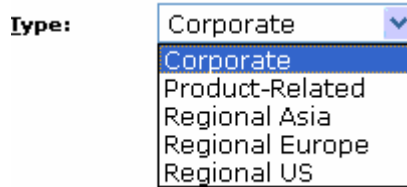
Figure 39: sys_DropDownSingle control

- ▪ sys_DropDownMultiple

Figure 40: sys_DropDownMultiple control

   The sys_DropDownMultiple control is similar to the sys_DropDownSingle control, but lets users choose multiple values from the predefined list. As with the sys_DropDownSingle control, the format of this control allows it to be compact, yet include a large number of options. Compare it to the sys_CheckBoxGroup, below, which also allows users to choose multiple values from a list of options but displays all options on the page.

- ▪ sys_RadioButtons
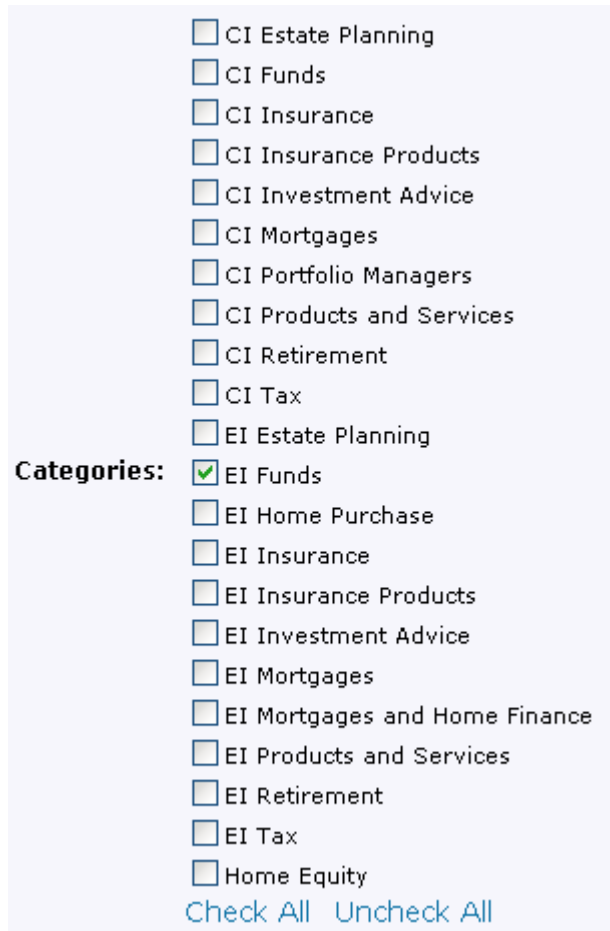
   The sys_RadioButtons control lets users choose one value from a predefined list. All values appear on the page with selection buttons preceding them. Selecting one option deselects all other options. The format of this control allows users to see all options at once, but if there are a large number of options, it may take up a lot of space in a Content Editor.
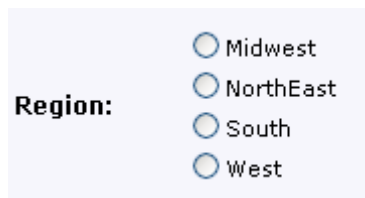
Figure 41: sys_RadioButtons Control

- sys_CheckBoxGroup

  The sys_CheckBoxGroup control lets users choose any number of the list of values shown. All values appear on the page with selection boxes preceding them. The advantage of presenting multiple choices using this control rather than a sys_DropDownMultiple control is that users have an easier time considering all of their options at once and are less likely to miss choosing an option. However, this control could potentially take up a lot of space in a Content Editor (although you can configure it in columns).

Figure 42: sys_CheckBoxGroup control

- sys_CheckBoxTree

    The sys_CheckBoxTree control differs from the sys_CheckBox in that it lets you display expanding main categories with sub-categories beneath them. The tree may expand to any level of choices, but categories with sub-categories beneath them are not selectable. Like the sys_CheckBox, this control lets users choose any number of selections preceded by check boxes.
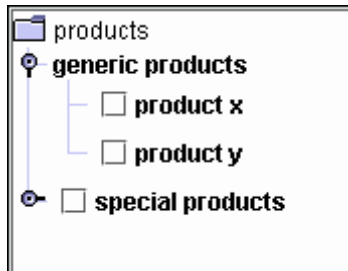
Figure 43: sys_CheckBoxTree control

Ally does not want the user to be able to select more than one category for each image. This eliminates the sys_DropDownMultiple, sys_CheckBoxGroup, and sys_CheckBoxTree controls. Since the categories for images may increase as the Site grows, Ally decides that the sys_RadioButtons control may begin to take up too much space on a Content Editor page. She decides to use the sys_DropDownSingle control, which will take up a small amount of space on the Content Editor page regardless of the number of choices it holds.

She decides to make the field optional, since some images may not fit into the categories chosen. Since she wants the categories to be brief in length, she chooses a data type of text and a size of 50 characters. The field has no special properties. Its basic properties are:

Name	Label	Control Name	Occur	Data Type	Format (size)
img_category	Image Category:	sys_DropDownSingle	optional	text	50

Event Content Type

The Event Content Type has three local fields, event_start, event_end, and event_type. It also includes the local child set field event_location, which has four local fields, event_city, event_state, event_address, and event_contact.

The event_start and event_end fields are easy for her to define. CM System offers only one date control, sys_CalendarSimple, so she uses it for both fields.

The sys_CalendarSimple control includes a text box for entering a date, and a calendar icon that the user can click on to open an enlarged calendar for choosing a date that is entered into the text box in the correct format.

Figure 44: sys_CalendarSimple control

Ally chooses a data type of datetime which does not require specifying a size. She makes the fields optional, since her company wants the ability to post events on the Web Site before a specific date is determined.

Name	Label	Control Name	Occur	Data Type	Format (size)
event_start	Event Start Date:	sys_CalendarSimple	optional	datetime	none

Name	Label	Control Name	Occur	Data Type	Format (size)
event_end	Event End Date:	sys_CalendarSimple	optional	datetime	none

The event_type field is intended to store the category of the event, and Ally defines it similarly to the img_category field. However, this field is intended to be used on Web pages, to explain to visitors what type of event is being advertised, and therefore requires a larger size.

Name	Label	Control Name	Occur	Data Type	Format (size)
event_type	Event Type:	sys_DropDownSingle	optional	text	255

Now Ally defines the properties of the fields in her event_location field set.

Noah explains that before defining the properties of the fields, she must make a few decisions about the event_location field set as a whole:

- First, she must decide if entering information into the field set is required or optional. If it is required, she must decide it there is a specific number of location entries that must be entered.
- Next, she must decide if users are allowed to reorder location entries after they enter them.

Ally wants to make sure that an event is not entered without a location, so she makes the event_location field set required. However, events in her company can take place at one or more locations, so she does not require a specific number of entries.

She decides to let users reorder the location rows after they enter them.

event_location
▪ required
▪ allow users to reorder entries

Now she can define the fields in the event_location field set.

The event_city and event_state are both short fields and she decides to use a simple sys_EditBox control and a data type/size of text/50.

Since the event_address and event_contact may require more text, such as several address lines, or contact phone and fax numbers, she decides to use sys_TextArea controls for these fields and give them each a data type/size of text/255.

Noah explains that she has the option of displaying fields in field sets on the main page of the Content Editor or only allowing contributors to view them when they enter or edit them in a separate child editor. Although she must enter all of the data into the field set fields in the child editor, she can display any number of them on the main page. The Rhythmyx Workbench term for displaying a field set field on the main page is *display in summary*.

Ally asks Noah what is the advantage of displaying or not displaying a field on the main page. Noah says that it is a matter of how much space the fields will take up on the main page and whether or not including them will help content contributors see what information is included in the Content Item or confuse them by presenting too much information at once.

Ally decides to display the city and state fields on the main page, but only allow content contributors to see the contact and address fields in the child editor.

Name	Label	Control Name	Occur	Data Type	Format (size)
event_city	Event City:	sys_EditBox	optional	text	50

Other properties
Display in summary

Name	Label	Control Name	Occur	Data Type	Format (size)
event_state	Event State:	sys_EditBox	optional	text	50

Other properties
Display in summary

Name	Label	Control Name	Occur	Data Type	Format (size)
event_address	Event Address:	sys_TextArea	optional	text	255

Name	Label	Control Name	Occur	Data Type	Format (size)
event_contact	Event Contact:	sys_TextArea	optional	text	255

The city and state fields will appear in the Content Editor as:
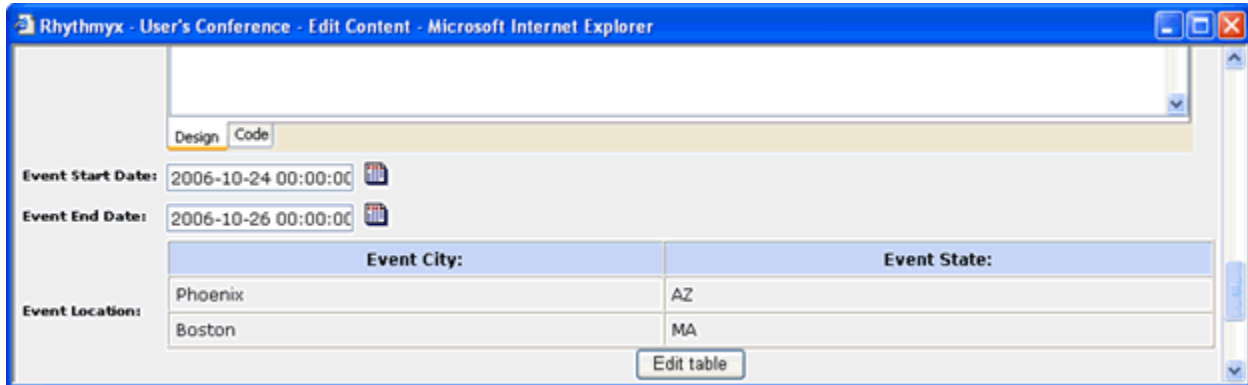
Figure 45: Event Location table

Before a contributor has entered data in the Event Location field set, the [**Edit table**] button on the Content Editor reads [**Add new item**]. After a contributor has entered data, the button reads [**Edit table**], as in the graphic above. The [**Edit Table**] button accesses a child editor for modifying existing entries and includes an[**Add new item**] button for adding new entries. In the child editor, an Action column includes controls for deleting the row ✕ , editing or viewing all of the fields in the row 🖉 , or changing the order of rows ⬍ . If users are not allowed to reorder rows after they enter them, the ⬍ does not appear.
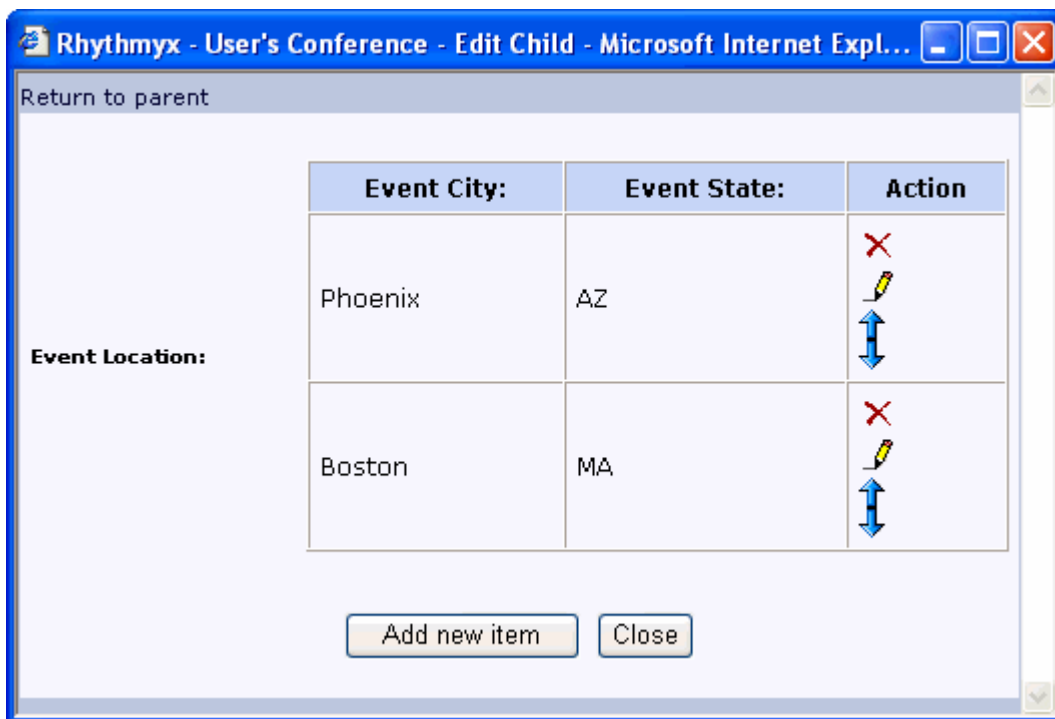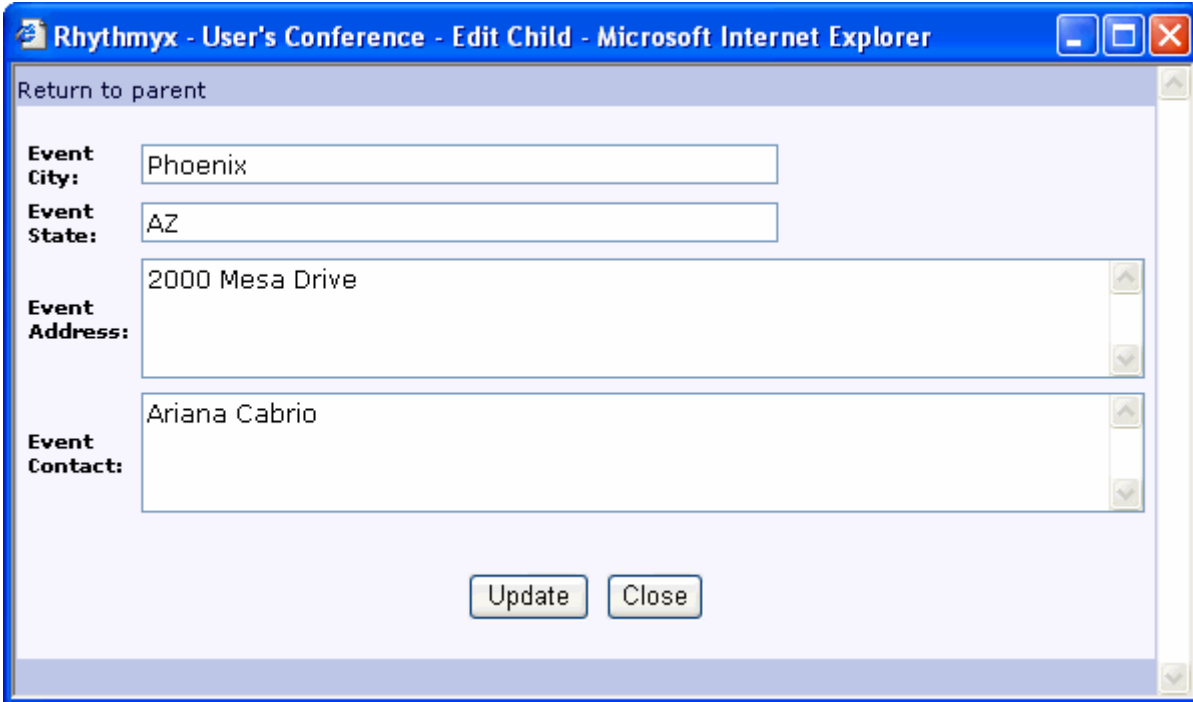
Figure 46: Child Table editor

To view or edit all of the fields in a row, the user clicks ✎ . This opens the child editor that is used to add or edit field set fields.

Figure 47: Child Table Editor, Add New Item page

To walk through a step by step example of how this control works, see The Event Content Editor.

NOTE: We have not covered the sys_VariantDropDown and sys_HTMLEditor controls. Although these controls are available in the Workbench, they are present for backwards compatibility and should not be used with new fields in CM System 6.0 and later.

Ally has now defined the fields in her shared field sets and Content Types. You can see a consolidated list of what she has specified in this section in the ***development plan*** (see page 117). Now she will proceed to design the global template and managed navigation for her Web Site.

Decomposing the Image Snippet and Other Snippets

Questions:
- What do you want the Image Content Type to do in addition to uploading and storing an image file (create and store a thumbnail of the image? store certain metadata about the image?)
- What formats do you want to give the Templates that can be shown in each Slot.

Ally understands that to display images on her Web Pages, she will need at least one Image Content Type. Ally wants to create a Snippet that simply displays an image. She will include this Snippet in Slots on the Home Page Template and any other Templates where she wants to display an image. The Image Snippet will be type-specific to the Image Content Type.

Figure 48: Images on Home Page

Noah, agrees that the Image Content Type is necessary and will appear in Slots on the top of the Home Page and in the center of the page. He explains that the image in the center of the page is uploaded and stored in an image Content Type like the others, but Ally must choose if she wants to include it in a Slot on the page or add it to a field that is represented by a rich text editor as an "inline Slot", for example:
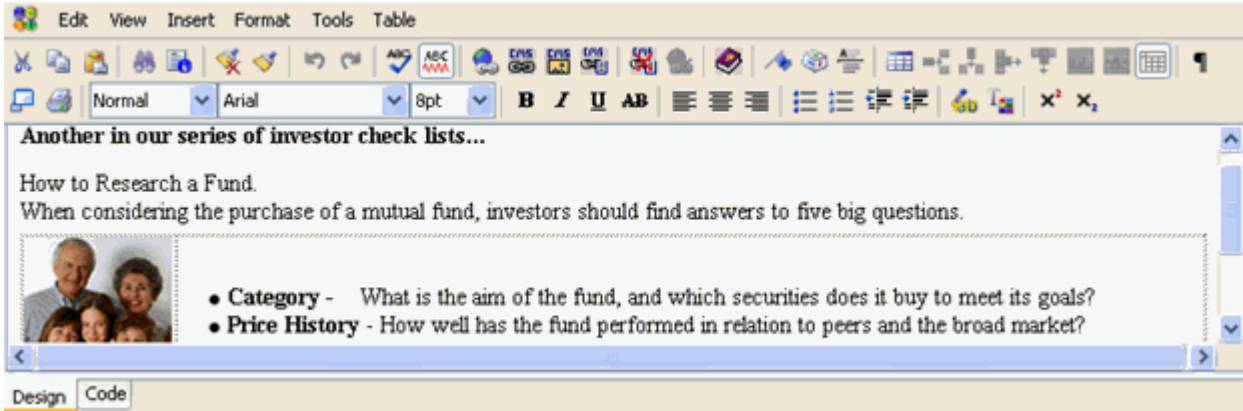
Figure 49: Graphic in Rich Text Editor

Ally decides that the central graphic could be entered as part of a field into a rich text editor, and she will use CM System's default inline image Slot.

How to decide if a graphic should be entered in a separate Slot on a page or in an inline Slot within a rich text editor:

Enter a graphic in a separate Slot if:

- You want to be able to display the text in the rich text editor without the graphic.
- You want to give the Web Master the option of choosing the graphic that appears beside the text in the rich text editor.
- The graphic is frequently updated, but the text in the rich text editor is not.
- You want the graphic to appear in a particular place on a page by inserting it into a certain normal Slot on a Template.

Enter a graphic in an inline Slot in a rich text editor if:

- You want the graphic to appear beside the text in the rich text editor wherever the text appears.
- You want the same graphic to always be associated with the text in the rich text editor.
- You want to be able to design unique formatting for the graphic and its text in the rich text editor.
- You want the content contributor to be able to determine where the image appears on each page.

Noah explains that in addition to the Snippet Template that assembles and displays the image, she will have to define a binary Template that retrieves the binary file. A binary Template does not include code for assembling Snippets or Pages; its function is to retrieve a binary file and its mime type from the Content Type fields that store them so the file is available to other Templates. The Snippet Template that Ally creates to display her image files must link to the binary Template in the code for assembling the Snippet.

NOTE: The procedure for configuring a binary Template is explained in the the topic "Implementing a Binary Template" in the *CM System Implementation Guide*. However, FastForward provides Templates that perform this function; you may find it easier to use these Templates than to create your own.

Earlier, Ally mentioned that she wanted the ability to store a thumbnail of the original image in the Image Content Type. Noah reminds her that the Content Type will need fields to store the original and thumbnail images, and certain fields required for displaying the images. In addition, the extension (java plugin) that uploads the image extracts additional information that she can optionally store in the Image Content Type. Noah will help Ally determine exactly what fields to include in the Image Content Type in the following modeling and design step.

Now Ally and Noah must go through all of the Snippets that are included in the Sidebar Slots, the List Slots, and the Breadcrumb Slots on the Web pages they have decomposed and perform the decomposition process on the Snippets to create Snippet Templates. Ally should note where she wants text or images on the Snippets to link to other pages. When the Snippets are implemented, the assembler source code must specify which fields are links and what pages they link to. For example, earlier we showed a graphic of content assembled in the Sidebar Slot:
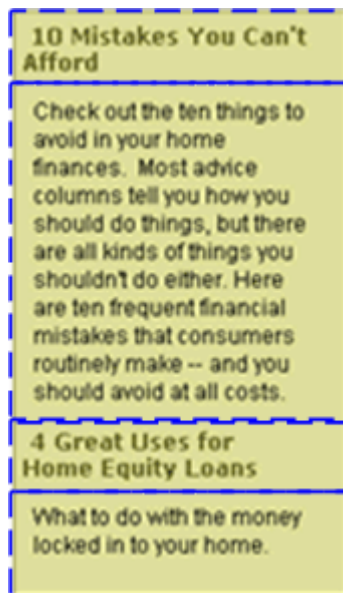
Figure 50: Snippet Template

In this case, the Sidebar Slot holds two Generic Content Items formatted using a Template that displays the Content Item displaytitle as a link to a page displaying the Content Item and the Content Item callout (marketing blurb) as plain text. When decomposing the Template, she should indicate which fields are links so that the implementer of the Snippet Template can code the field to link to the appropriate page. For example:
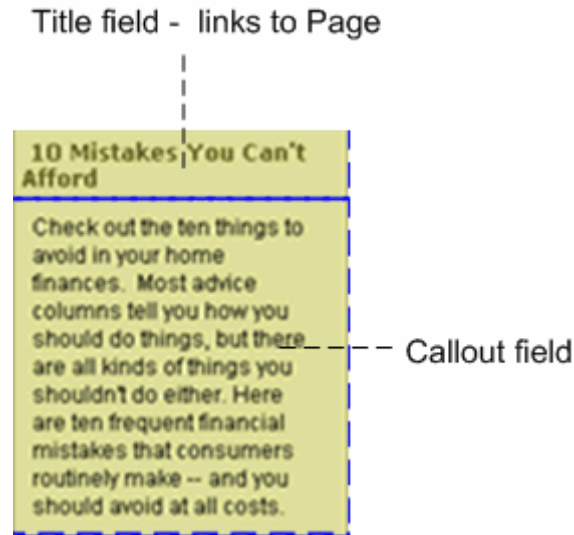
Figure 51: Snippet Template decomposed

We will not repeat Ally and Noah's decomposition process for each Snippet Template, since it is similar; however, it is important that Ally attempts to plan every Template and Slot on the system before implementation begins.

Ally must also specify which Templates each Slot can hold. She will begin with the Templates she currently sees in the Slots on her Web Pages, but may choose to add others as options.

C H A P T E R   6

Designing the Global Template and Managed Navigation

Questions:
- What do you want the look and feel of your Web Site to be?
- What do you want the navigation components of your Web pages to include?

When Ally began looking at the pages in her Web Site to initially assess her system's Content Types, Noah had her cut out the uniform content bordering each page. This area (as well as portions of the formatting applied to content within each page) composes the Global Template.

The Global Template is applied to a set of pages in a Site, all the content in a site folder, or even just the content that uses a specific Template. In Ally's Web Site, as is often the case, most of the pages in the Site will include the Global Template to maintain a look of uniformity.

Noah suggests that they begin examining the Global Template by looking at a complete page that includes it. In the following graphic of the About Enterprise Investments Page, the cut out outer portion of the page outlined in black is referred to as the Global Template. In this case, the Global Template takes the form of what is commonly referred to as a C-clamp around the page, and consists largely of navigation links. The Global Template determines the look and feel of the C-clamp, for example, the green background, the banner and the font style. In addition, its cascading style sheet can determine aspects of the look and feel of portions of the page. The green sidebar on the inner portion of the page would probably be determined by the Global Template's css so that it matches the C-clamp surrounding the page.
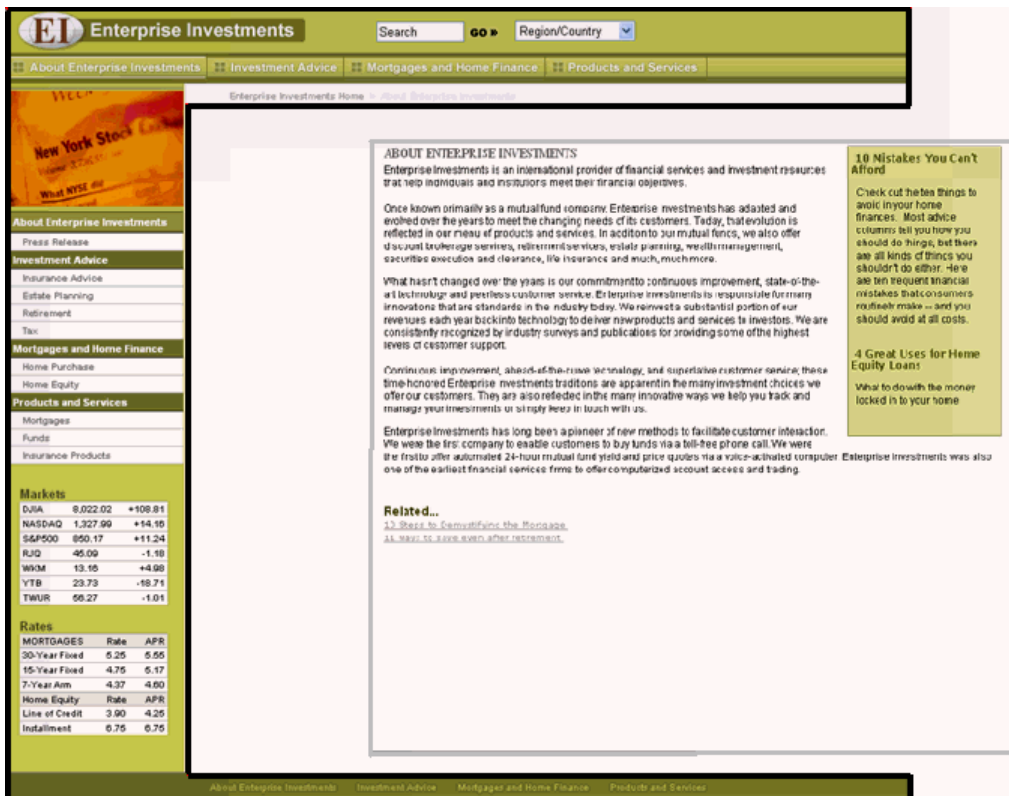
Figure 52: In this graphic, the Global Template is outlined in black

Frequently, the Home Page of a Web Site has different characteristics than the Site's other pages and uses its own specialized template instead of the Global Template. Note that the Home Page Ally's Web Site uses a separate Template that incorporates much of the code in the Global Template:

Figure 53: Ally's Home Page

Choosing the Web Pages' Look and Feel

Noah begins by asking Ally if she wants to maintain the current appearance of her Web Site. Ally would like, if possible, to use most of the same graphics for the logo, and button designs, and to apply a similar range of green and brown shades to her navigation panels, content pages, and text. Noah explains that the look and feel is usually specified in one or more stylesheets that are referenced in the HTML associated with the Global Template.

Ally already has cascading style sheets that define some of the look and feel styles that she has been using in the Web Site. They can make any necessary modifications during implementation and reference the stylesheets in the HTML they develop for the Global Template.

Designing the Site's Managed Navigation

Questions:
- Do you want to use text or images as your navigation links?
- Do you want to make all pages on the Site available in your left navigation pane, or only section pages?
- Do you want sections in the left navigation pane to be collapsible?
- Do you want to include breadcrumbs?

Noah explains that Global Templates usually include the Managed Navigation portion of the Web Page. In the C-clamp structure that Ally is using, Managed Navigation refers to the links at the top of the page, the left side bar, and the bottom of the page, that appear identically, or in a similar pattern on each page. In addition the breadcrumbs at the top of each page are an aspect of Managed Navigation (breadcrumbs are links showing the path of pages that the visitor has followed). The implementer or Web Master manages these navigation links in CM System so that CM System automatically generates them when assembling a page. The Global Template includes them in navigation Slots. Ally must review with Noah the links that she wants to appear in the different areas of her Global Template, and make sure that CM System can produce them using Managed Navigation.

At the top and bottom of her page, Ally's navigation bars include four links to the same section pages. She would like to maintain these basic links. The top navigation pane consists of buttons that are actually graphics that include text. Although Ally likes the graphics, she is considering changing them to text menus that expand to include the options shown in the left navigation bar. For example, the Investment Advice link would be a drop menu that included *Insurance Advice*, *Estate Planning, Retirement,* and *Tax*. Noah suggests that Ally avoid using the drop menus. Although she may see Web Sites where the top navigation bar includes drop menus, this style has become less popular because it makes the navigation area too complex.

Ally agrees and returns to the idea of using buttons, which she likes because they lend a more professional appearance to the Web Site. She would also like to create popout versions of each button graphic that appear when the button is clicked. Noah presents another caveat: in general css-styled text works better than graphics for navigation links because the text is readable by screen readers and therefore meets accessibility standards, and scaling graphics to the correct size is less precise than specifying measurements for text areas in a cascading style sheet. However, Ally decides to use the graphics since visitors to the site are familiar with them, and her Web designers have not had a problem displaying them in the past.

Figure 54: Top Navigation Bar

The bottom navigation bar includes simple text links to the same pages. Ally would like to maintain them, since Web Site visitors may scroll down and lose access to the top navigation buttons.

Figure 55: Bottom Navigation bar

Ally has breadcrumb links on her Web pages directly below the top navigation bar. Currently, if a visitor to the Site had begun at the Home Page, then linked to the Products and Services page, and then linked to the Funds page, the breadcrumbs would appear as:

Enterprise Investments Home > Products and Services > Funds

Figure 56: Breadcrumbs

Noah lists some of the other options for displaying breadcrumbs and asks Ally to consider if she wants to make any of these changes:

- Do not list the Home page since users always have alternate ways to return to it.
- Do not list the current page since users are already looking at it.
- Connect the links with another character or characters than ">". For example "..." or ">>" or a graphic such as ▷ .

Ally feels that her current method is the most common she has seen on Web Sites, and wants to keep it since users will easily understand its function.

Now, Ally and Noah look at the left navigation bar:
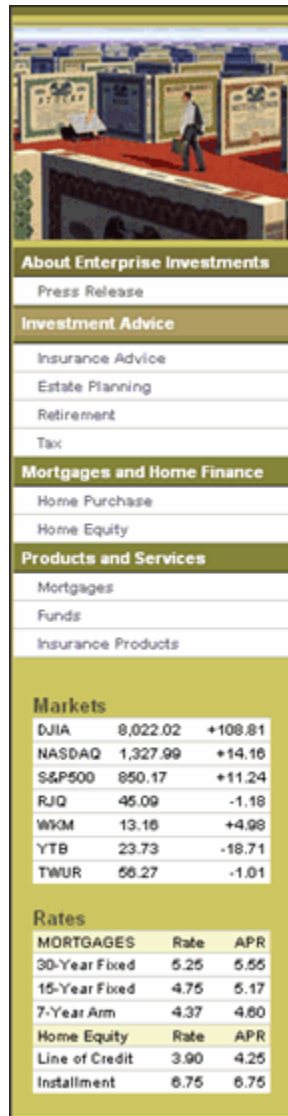
Figure 57: Left Navigation Bar

The left navigation bar includes a graphic at the top that changes depending on which section of the Site the visitor has entered. Below the graphic are the same four main navigation links that appear on the top and bottom navigation bars. But the left navigation bar includes secondary page links below each main page link. Below all of the links are tables showing market values and mortgage rates.

If she clicks one of the links for the main pages, it appears darker green. If she clicks one of the links for the secondary pages, it appears yellow.

Ally wants the links to the four main sections in her Site to remain as they are, but asks Noah what variations are common in listing the secondary pages. Noah offers some common variations on Ally's structure:

- She could include all of the pages in the Site in the left navigation bar, inserting some at a third or even a fourth level below the secondary pages.

- She could allow the main navigation links to be collapsible. The visitor to the page could either click a main link to show the sub-links below it and collapse all of the other main links, or click a control to expand and collapse each section.

Ally decides that these features are not really necessary. She provides Noah with the HTML for what is currently the common portion of her Web pages because they can use this to develop the Global Template.

Planning Site Folders for Managed Navigation

Although it is during the implementation process that the correct Site Folder structure is set up so that CM System can automatically achieve the Managed Navigation that Ally has planned, Noah tells Ally that she should begin to plan out her Site Folder structure to be sure her she can set up her links the way she wants to in CM System.

He gives her a brief explanation of the way CM System automates Managed Navigation. The root Site Folder contains a special navigation Content Item called a NavTree. The NavTree includes a Slot that includes the main menu items that appear in each of the navigation bars. For example, for Ally's site, the root folder NavTree would include *About Enterprise Investments*, *Investment Advice*, *Mortgages and Home Finance*, and *Products and Services*. Site Folders within the root folder are associated with these main menu items. Each contains a navigation Content Item called a Navon that includes:

- text and optionally an image to use as a link to the main menu items
- the name of the page that the link accesses
- a Slot for including the submenu items that make up the listings of secondary pages in the left navigation bar

In CM System Managed Navigation, the folder holding the Navon also includes subfolders for the submenu items in the listings of secondary pages, and typically includes the Content Item (page) that the link in the Navon accesses, although this item may be present in another folder. Each subfolder may also include a Navon that includes a link to the submenu item and typically includes the sub-Content Item also. The Global Template includes code that automatically propagates its navigation Slots with all of the navigation links specified in the Navons.

NOTE: This is a very abbreviated explanation of how Managed Navigation works in CM System. For a full explanation, see the chapters "Setting Up the Publishing Site and Basic Navigation" and "Managed Navigation" in the CM System Implementation Guide.

Together Ally and Noah put together the following preliminary Site structure:
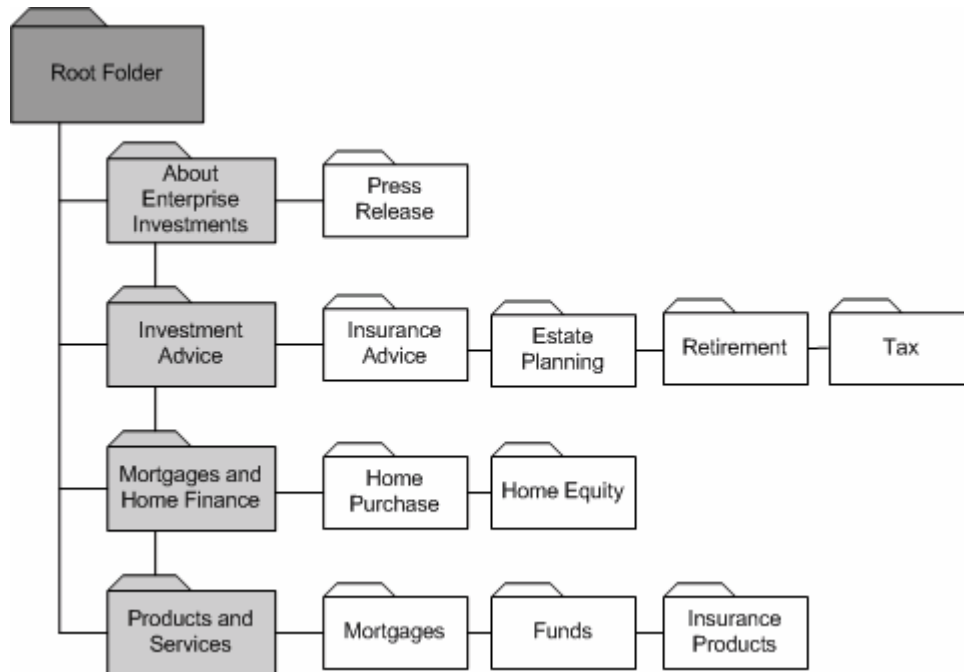
Figure 58: Site Folders for Managed Navigation

This site structure shows the folders that will hold Managed Navigation elements. During implementation, Ally will also include other folders for storing Content Items that do not appear in navigation elements of the Site.

Now that Ally and Noah have planned her Global Template and Managed Navigation, they can begin to sketch out the Roles, Communities, and Workflows for her system.

C H A P T E R   7

Planning Roles, Communities, and Workflows

Noah suggests that now Ally focus on sketching out the Roles, Communities, and Workflows that her system requires. He describes these elements as forming the foundation of her development environment because without them, she could not assign individual users access to certain Content Items and functions to perform on them. Additionally, she could not apply her company's operating rules to the process of creating and approving content. He briefly defines these three elements for her:

- Role - a collection of users associated with the same permissions and access in CM System. For example, users in the same Role can access the same Sites and Content Types, and perform the same functions on Content Items.

- Community - a Role or group of Roles with access to similar information in CM System. A Community is given access to CM System components such as Content Types, Display Formats, Menus, Searches, Sites, Templates, Views, and Workflows. When a Role is associated with a Community, it then has access to the components that the Community has access to.

- Workflow - The business process that a Content Item progresses through during its lifetime. In CM System this process involves the Content Item moving through a progression of States in which users in different Roles approve or reject it as it is prepared for publishing. Usually a Content Item eventually reaches a public State. Once the Content Item is in a public State the system can publish it to a Site.

  In CM System, the movement of Content Items between States is referred to as Transitions. Implementers can configure Transitions to send emails to specific Roles or Users when a Transition occurs. These emails are referred to as Notifications.

Defining the Workflow Process for Content Types

Noah suggests they begin by looking at Ally's current approval (Workflow) process for each of her Content Types. Ally explains that it is somewhat loosely defined, and follows one of two patterns.

Usually the process is as follows:

Either Writers or Financial Consultants create initial content. Writers send their content to copy editors, and financial consultants send their content to financial editors. The editors review the content and make necessary changes. The editor either sends it back to the creator with comments and starts the process over, or makes some edits and sends it to a quality assurance specialist who determines that it will appear correctly as a formatted page on the Web Site. The quality assurance specialist may send it back to the editor for review if extensive changes are required, although the editor rarely objects, since the text content always stays the same. If the quality assurance specialist does not require changes to the item, he or she passes it on to a Web Administrator who has the ability to further modify its format for the Web Site and then publish it. The Web Administrator can also unpublish the content and either archive it, or send it back to any earlier State. In addition, implementers have the ability to create content and move it through any of the Workflow States described above.

However, on occasion, Web Administrators create specifications for content that is necessary for the Site's organization. Usually they require this content as soon as possible. In these cases, the Writers or Financial Consultants who create the content do not have editors and quality assurance specialists review it, and send it directly to Web Administrators, who then proceed as in the above Workflow.

Noah suggests that they sketch out a CM System-style Workflow diagram that shows the States and Transitions in the first Workflow that Ally has described. They will add the Roles that can perform each Transition below the name of the Transition. After finalizing this Workflow, they will return to the other one, which is simpler.

Note: For this chapter, we have developed graphics of Workflows using the CM System Workflow Editor. In an actual modeling and design session, the Workflow would probably be sketched on a whiteboard.
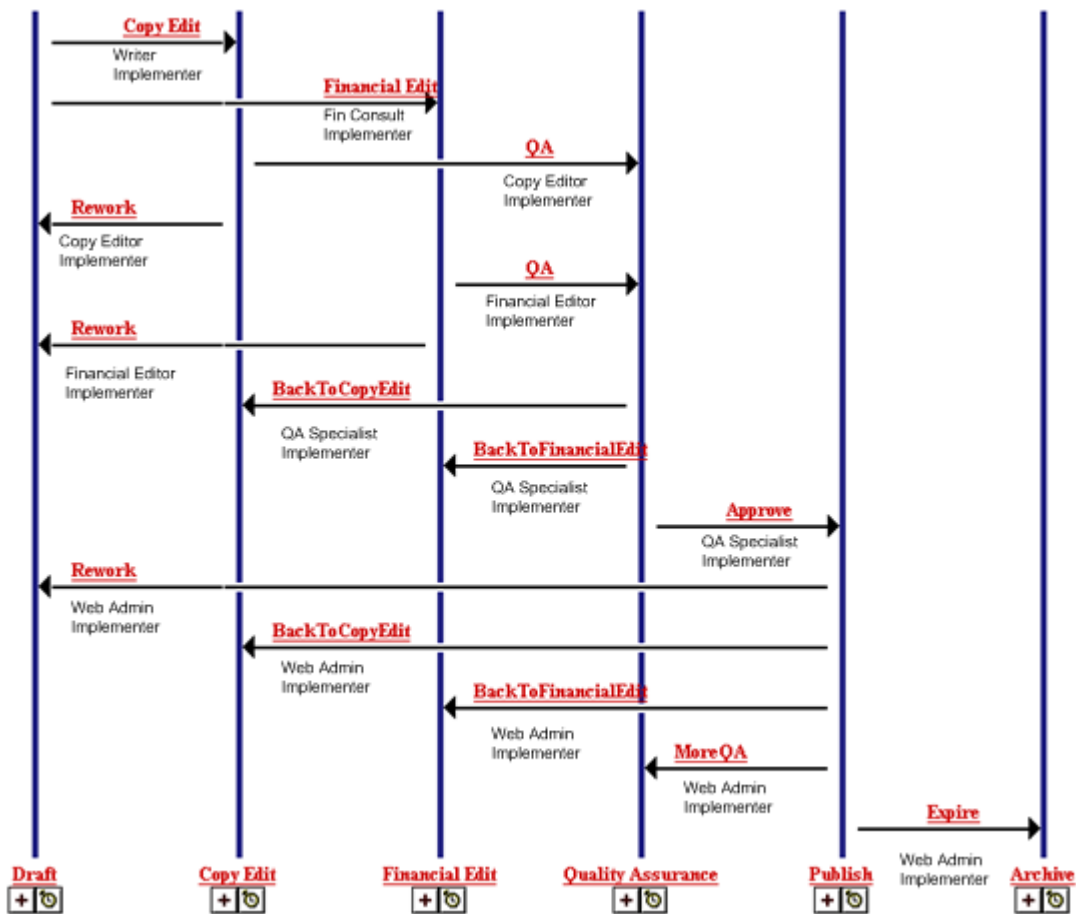
Figure 59: Ally's Original Workflow

Noah says that now they should see if they can simplify the Workflow in any way so that the process of creating content and getting it onto the Web Site is more streamlined. He suggests that they begin by looking at the Roles assigned to each Transition. Currently in the Workflow, users are transitioning content according to their role in the company. He suggests that instead Ally think of Roles in terms of Workflow functions. For example, he says that it is unimportant if a Financial Consultant or a Writer creates a Content Item. In terms of the Workflow, both of these users can be assigned to an Author Role.

Noah mentions that it is necessary to have one Role be a Workflow administrator. This Role must be able to perform transitions in all States and check in any content item to prevent content from becoming "stuck" in a Workflow. He notes that Ally really already has this Role, which she calls the Implementer. Ally changes the name of the Implementer Role to Administrator to better reflect the Role's function in the Workflow.

Noah explains that the feature of Ad Hoc assignment enables them to merge copy editors and financial editors into the same Editor Role, by allowing the user Transitioning the item to choose a specific individual assigned to the "to" State to act on the Content Item. In this case the user who performs the Transition could use Ad Hoc assignment to choose the correct user, either a copy editor or financial editor. Noah notes that combining these Roles would further simplify the Workflow by eliminating the need for both a Copy Edit and a Financial Edit State. Instead they can have a single Review State. He also mentions another option. If Ally wants to merge the two states into one Review State, but maintain two separate Roles, she can use another Workflow feature that enables her to specify a required number of approvals before a Content Item Transitions, and to name the Roles that must perform the approvals. In this case, she could require two approvals, one performed by a Copy Editor and one performed by a Financial Editor.

Ally agrees to merge the two Roles and States, but decides that in this case using Ad Hoc assignment or requiring two specific Roles is unnecessary; the distinction between the two editors is really blurred, and both can perform a satisfactory edit on any Content Item.

Noah asks Ally how necessary she feels it is to include the Quality Assurance State. She has suggested that the purpose of this State is to allow the Quality Assurance specialist to confirm that the content's page format will appear correctly, and notes that Web Administrators perform the same task. Noah suggests that this review can be performed while the Content Items are in the Review State, so they can remove the Quality Assurance State. Ally agrees, and says further that the Web Administrator is better suited to perform this review. Noah reminds Ally that she can require approvals from specific Roles for the transition to Public; in this case, she might want to require approvals from both the Editor Role and the Web Administrator Role. Ally says she prefers this option. (NOTE: The Standard Workflow provided with FastForward does not use this technique, but we will demonstrate how to implement it in the *CM System Implementation Guide*.)

With these revisions, Ally's Workflow is more streamlined and appears as:

Figure 60: Ally's Streamlined Workflow

Noah suggests that she add a few more features in this Workflow that have proven useful to customers:

- A Direct to Public Transition for items in the Draft State. This Transition should only be available to Web Administrators and Administrators. It enables the Web Administrator or Administrator to move crucial content onto the Web Site immediately.

- A Quick Edit State for items in the Public State. If a published item needs to be quickly revised and republished, users in assigned Roles can send it to the Quick Edit State, edit it, and then immediately transition it back to the Public State. The Quick Edit State is a special State that makes it easier to process a minor change to a published item without sending it through the entire Workflow again. CM System provides a special Publishable value, *Ignore*, for this State. When a Content Item is in a State with this Publishable, its current status is ignored and CM System processes it based on it processing in the previous publishing run. If the Content Item was public, the last Public Revision is published. If it was not public, the Content Item is not published.
- A Pending State where Content Items can be held after completing Review and waiting for their Start Date to pass so they can become Public.

Ally agrees that these features would be beneficial and adds them to her Workflow.

Noah then recommends a few Aging Transitions. Aging Transitions are Transitions that CM System automatically performs on a Content Item. Aging Transitions can occur:

- repeatedly after set intervals;
- once after a set interval; or
- once after a date specified within the Content Item has passed.

An Aging Transition does not necessarily move the Content Item to a new State. Repeating Transitions in particular often keep the Content Item in its current State. (Such a Transition is often referred to as a "circular Transition".) Such Transitions are often used to send Notifications to users that some action is required on a Content Item. Noah suggests adding the following Aging Transitions:

- A date field Transition that automatically Transitions the Content Item from Pending to Public when it reaches its Start Date.
- A date field Transition that sends a Notification reminding the assigned Roles that the Public Content Item is reaching its Expiration Date.
- A date field Transition that automatically Transitions the Content Item from Public to Archive when it reaches its Expiration Date.

Since Noah has recommended an Aging Transitions that generates a Notification, Noah and Ally discuss Notifications in more detail. Noah explains that a Notification is a Template used to automatically generate an e-mail when a Transition occurs. Ally decides to send emails to Roles in the Archive State after both Age to Archive and Expire transitions, since these users often want to revise archived content and send it back to the Web Site.

Noah points out that all of the Notifications Ally has specified go to the users in Roles assigned to the State that the item is transitioned *to*, which is common. He notes that she could also have chosen to send notifications to users that the item is transitioned *from*, or to individually specified email addresses.

At this point, Noah suggests that they re-sketch her Workflow with their latest additions and changes. He sees some additional changes that he wants to suggest to Ally, but wants to make sure they are keeping track of the changes they have already made.
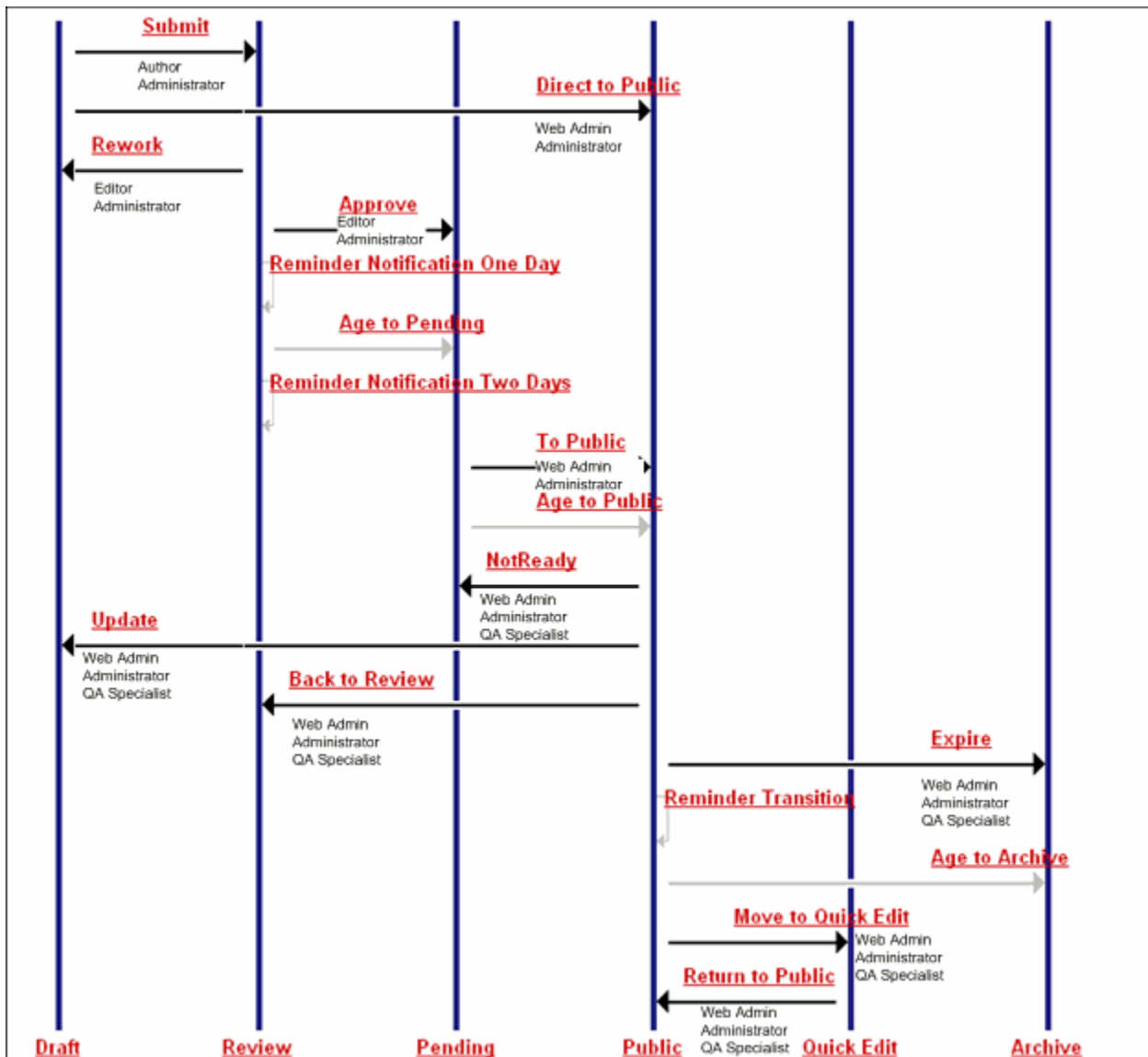

Figure 61: Ally's Enhanced Workflow

Ally has added Notifications to the following Transitions:

- Submit (to "to" Roles)
- Review, (to "to" Roles)
- Public (to "to" Roles)
- Expire to Archive (to "to" Roles)
- Age to Archive (to "to" Roles)
- Rework (to "to" Roles)

Noah makes a few suggestions regarding the current iteration of the Workflow. Now that Ally has the Move to Quick Edit and Return to Public Transitions, her other Transitions that remove Content Items from the Public State for editing are superseded since they are less efficient. Ally agrees to remove the Back to Review, NotReady, and Update Transitions.

He also notes that she made a point of sending Notifications to Roles in the Archive State, but does not include any Transitions from the Archive State. Ally says that she has seen Web Administrators publish content right back to the Web Site after it has been archived, so she should add that Transition. She has also seen them send content back to Authors for updating, so she will add that Transition.

Ally also has some ideas for enhancements. Noah previously explained the Ad Hoc assignment feature to her. She understands that if she makes a Role an Ad Hoc assignee, a user can transition a Content Item to the State and assign a user as an Ad Hoc assignee so that only that user can work on the Content Item. Ally decides to make the Author Role in the Draft State an Ad Hoc assignee so that other Roles can send Content Items back to the users who created them. She would also like the Editor who sends back the content to explain to the Author what changes to make in the Notification, but is not sure if users can customize Notifications in this way. Noah explains that CM System can add approver comments to Notifications by inserting a comment made during the Transition into the email sent. Ally can require that a comment is included with the Transition to ensure that the "to" Role receives instructions.

Ally wants to add an additional change. She decides that the Web Admin Role should have the same ability that the Administrator Role has to perform most Transitions. She gives the Web Admin Role the ability to perform the Submit, Rework, and Approve Transitions.

Noah has also discussed the option of requiring certain numbers of approvals as well as requiring certain Roles to perform the approvals. Ally decides to require that the Editor and Web Admin Roles perform the Approve Transition before a Content Item can move to the Pending State.

Her final Workflow appears as:
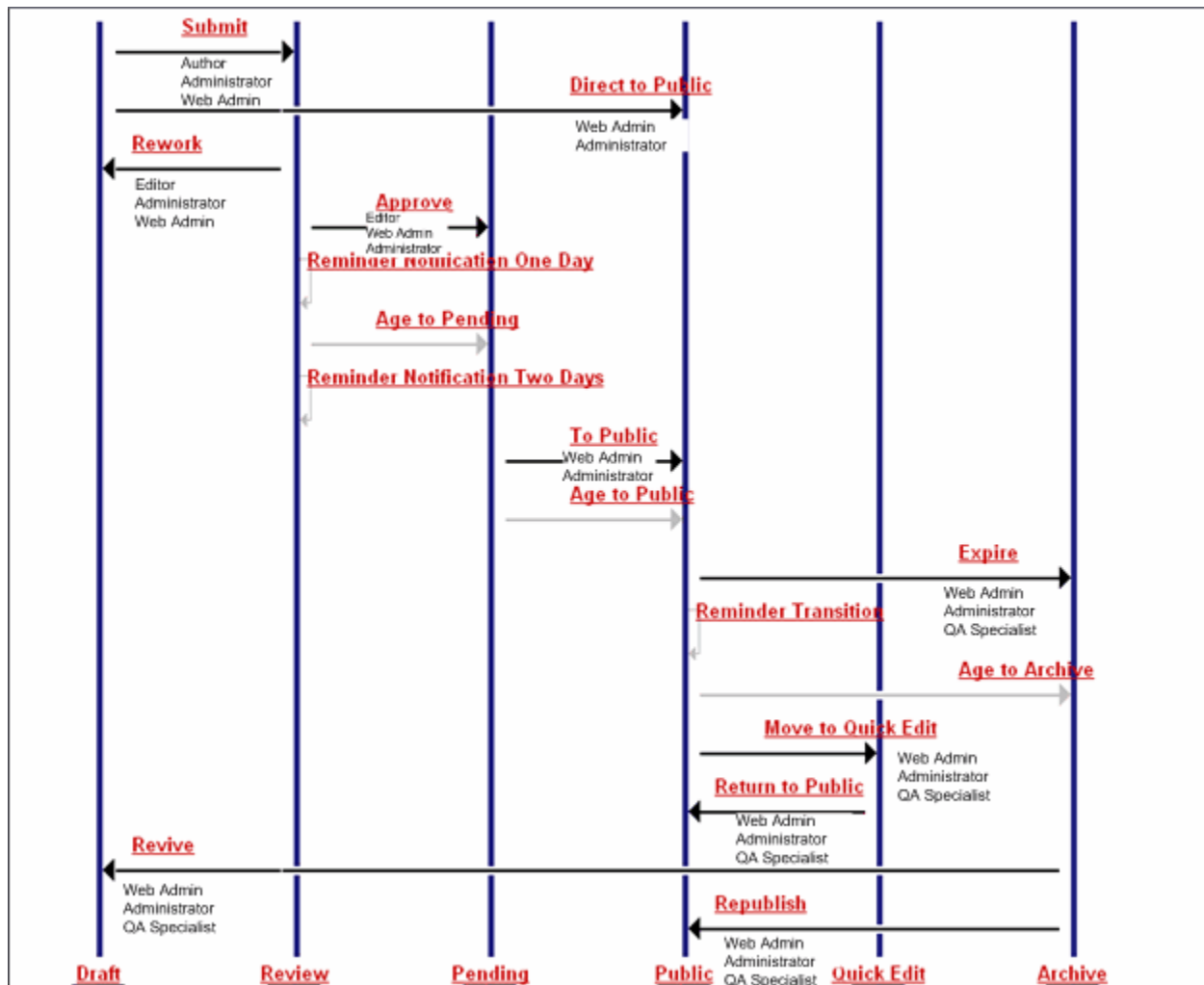
Figure 62: Ally's Final Workflow

Noah says they must plan out two more details of the Workflow before it is ready to be implemented. First, each time a Role is assigned to a State, its access to Content Items in that State is specified. Options include:

- assignee

  Users in Roles with assignee access have full control over the Content Items in that State. They can check out the Content Items, edit them, and Transition them to another state. A Role must have assignee access to permit its members to have any ability to act on Content Items.

- reader

  Users in Roles with reader access can see Content Items when they are in the State, and can open them in read-only mode. They cannot perform any operation on the Content Items.

- none

  Users in Roles with none access cannot see Content Items in the State, and cannot act on them.

Ally decides to make the Author Role a reader in all States other than the Draft State, in which it is an assignee, so the author of content always has the ability to view the content that he or she has created. Similarly, she decides to make the Editor Role a reader in all States other than the Review State, in which it is an assignee, and the Archive State, in which it would have little interest, so the editor of content always has the ability to view content that he or she has edited. She also decides to make the QA Role a reader in the Review and Pending States, since the Role can act on content in some States surrounding it, and may have an interest in viewing it in these States. Noah adds one more Role as a reader to the Public State. He explains that the RxPublisher Role, an internal CM System Role, must have reader access to content in this State in order to Publish it. To summarize, Ally's reader Roles for the Standard Workflow are:

- in Draft State - Editor Role
- in Review State - Author and QA Roles
- in Pending State - Author, Editor, and QA Roles
- in Publish State - Author, Editor, and RxPublisher Roles
- in Quick Edit State - Author and Editor Roles
- in Archive State - Author Role

The second detail that remains to be resolved is to determine whether Default Transitions need to be assigned for any States, and if so, which Transitions should be specified as Default. Default Transitions are used in Relationship processing, which sometimes automatically Transitions Content Items from one State to another. In the standard implementation of CM System, the Active Assembly - Mandatory and Translation - Mandatory Relationships typically move Content Items from Pending to Public when their related Content item makes the same Transition, while the Promotable Version Relationship moves the current Version of the Content Item to Archive when the Promotable Version Transitions to Public. Note that if no Default Transition is defined, the processing uses the first Transition in alphabetical order by name. As this can result in undesired behavior, best practice requires that Default Transitions be designated from the Pending and Public States. Aging Transitions cannot be designated as Default Transitions, so the decision for the Pending State is easy. The only option is the To Public Transition. The Public State is somewhat more complicated, as two options are available: the Expire Transition to Archive and the Move to Quick Edit Transition to Quick Edit. Since the desired behavior of the Promotable Version is to move the current Version to Archive, the logical choice is the Expire Transition.

Since this is the Workflow assigned to most content, Ally calls it the Standard Workflow.

Noah says that now it should be easy for them to design the second Workflow that Ally described. All they have to do is remove the Review State, which will also remove the special aging Transitions associated with the State. He asks Ally if approvers in the Pending State ever send content back to Authors during this type of Workflow. She explains that since the purpose of the Workflow is to get the content published quickly, approvers in the Pending State make any necessary changes themselves and then publish the Content. Since this Workflow is much simpler than the Standard one, she decides it is not necessary to make the Author Role in the Draft State an Ad Hoc assignee.

They easily sketch out this Workflow, which is an abbreviated version of the Standard Workflow, and name it the Simple Workflow.
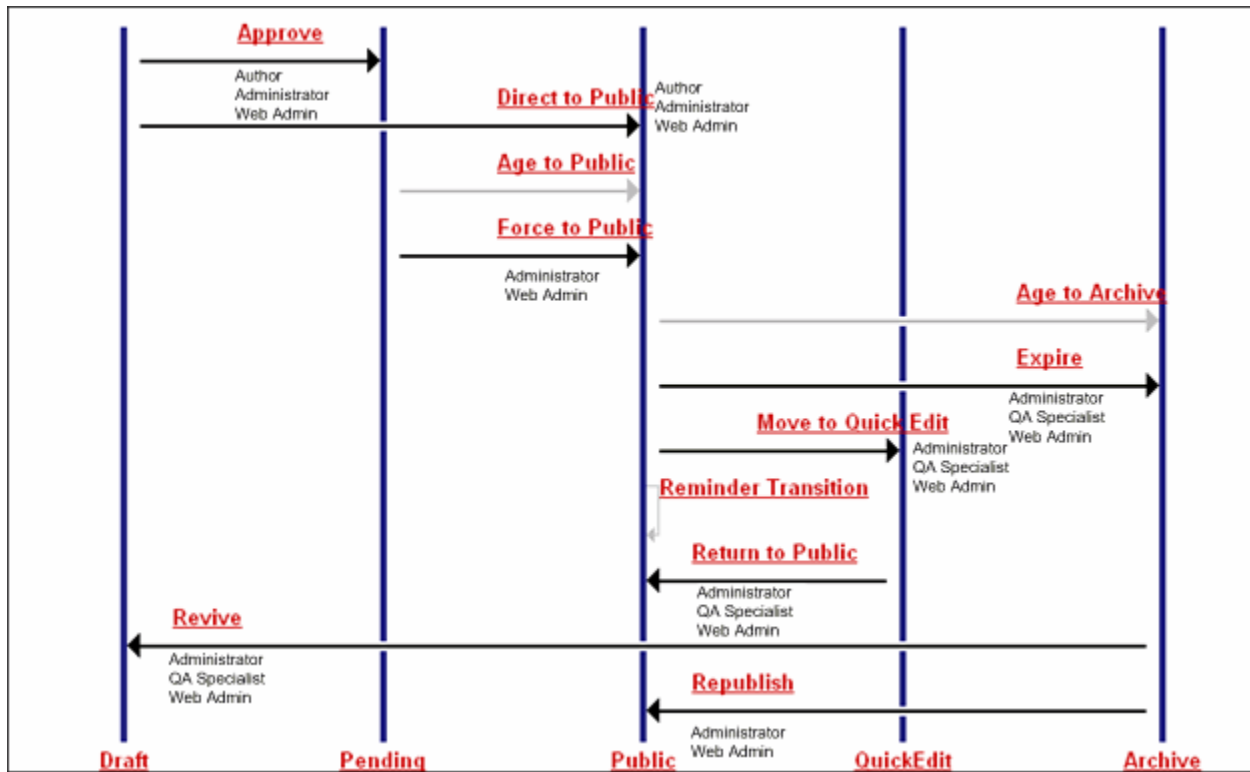
Figure 63: Ally's Simple Workflow

Ally's reader Roles for the Simple Workflow are the same as those for the Standard Workflow with the exception of those Roles that are readers in the Standard Workflow's Review State.

Now Ally has finished defining her Workflows, and Noah and Ally are ready to begin discussing her Communities.

NOTE: CM System comes with a Simple Workflow and a Standard Workflow that are nearly the same as Ally's Simple and Standard Workflow. Many customers are able to create their Workflows by creating variations of these two FastForward Workflows. See Workflows for more information about how to implement one Workflow by modifying another one.

Designing Communities

Noah has already stated that Communities define a Role or a group of Roles that require access to similar components in CM System. He now adds that each Site usually has at least one Community, but often has more if users who work on the Site require access to different Content Types, Templates, Workflows, etc. He recommends that Ally keep her Communities simple; their main purpose is not security, but to simplify what users see in CM System so they do not have to view components they never use and probably do not understand. In the future, she can add additional Communities if necessary.

Ally suggests that she could begin by creating an Enterprise Investments Admin Community that has access to everything, and then create a basic Enterprise Investments Community that has access to all components except those used for Site navigation, such as navigation Content Types and Global Templates.

Noah feels that this is a good plan. Viewing navigation and Global Template components could confuse typical content contributors. Although they may not use certain other components (for example, a writer might not create an Image Content Item) it is unnecessary to complicate the design of Communities by attempting to guess what each type of user will need to access.

Planning Community and Workflow Roles

Previously, Noah defined Roles as collections of users with the same access and permissions in CM System. He now explains that although all Roles are defined in the same way, it is considered a best practice to define "Workflow" Roles and "Community" Roles separately.

Noah says that Roles associated with Communities function to give users access to the CM System components such as Content Types, Display Formats, Menus, Searches, Sites, Templates, and Views that are associated with Communities. Roles associated with Workflows function to give users reader or assignee access Content Items in a Workflow State. As Ally already knows, Roles with reader access can view the Content Item in the specific State, while Roles with assignee access can Transition the Content Item to another Workflow State.

The purpose of separating Workflow and Community Roles is to allow a user in one Workflow Role to have the ability to perform a certain Workflow Transition in multiple Communities. The user has to be in a Workflow Role that has permission to perform the Transition, and a Role associated with the Community of the Content Type. It is easier if these associations are separated into two Roles.

For example, suppose Ally's system had two Communities representing two different sites: Enterprise Investments and Corporate Investments (this is actually the case in FastForward). If she did not separate Community and Workflow Roles, for each Community, she would have to create all of the possible Workflow Roles. By separating Community and Workflow Roles, she can create one set of Workflow Roles, and assign users to a Workflow Role and either one or both Communities depending on which Community content she wants the users to have access to. For example, instead of:

- Corporate Investments Author Role
- Corporate Investments Administrator Role
- Corporate Investments Editor Role
- Corporate Investments Web Admin Role
- Corporate Investments QA Specialist Role
- Enterprise Investments Author Role
- Enterprise Investments Administrator Role
- Enterprise Investments Editor Role
- Enterprise Investments Web Admin Role
- Enterprise Investments QA Specialist Role

She can simply create:

- Community Roles
    - Corporate Investments Role
    - Corporate Investments Admin Role
    - Enterprise Investments Role
    - Enterprise Investments Admin Role
- Workflow Functional Roles
    - Author Role

- Administrator Role

- Editor Role

- Web Admin Role

- QA Specialist Role

Ally still has the option of creating more restrictive Community Roles. For example, if Ally wanted to limit the access of the Web Admin and QA Roles to their specific Communities, she could create the Community-specific Roles:

- Corporate Investments Web Admin Role

- Corporate Investments QA Specialist Role

- Enterprise Investments Web Admin Role

- Enterprise Investments QA Specialist Role

While users in a generic Web Admin or QA Role would always have the highest access available to those Roles regardless of the Content Item's Community, users in Community-specific functional Roles only have the access available for Content Items within that Community. For example, suppose Lisa Kerr is a Member of the following Roles with the specified access:

- Corporate Investments Web Admin

  - Review: read

  - Pending: assignee

- Enterprise Investments QA

  - Review: assignee

  - Pending: read

If Lisa opened a Content Item from the Corporate Investments Community, and that Content Item was in the Review State, she would be able to see it and examine its properties, but could not act on it. If another user Transitioned the Content Item to the Pending State, Lisa would have full control over it. If she opened a Content Item in the Enterprise Investments Community and it was in the Review State, she would have full control, including the ability to Transition the Content Item to the Pending State. Once that Transition was complete, however, Lisa would have only read access to the Content Item; she would be able to see the Content Item and to review its properties, but she would not be able to act on the Content Item.

Ally would have to be careful when assigning users to their Roles, however, because users are granted the highest access. If, in addition to being a member of the Roles listed above, List was also a Member of the generic Edit Role, she would have the highest access granted by these three Roles. If she opened a Content Item in the Corporate Investments Community, she would have full control in both the Review and Pending States. In the Review State, The Editor Role would give her the highest access (assignee), while in the pending State, the Corporate Investments Web Admin Role would give her assignee access in the Pending State.

After discussing her needs with Noah, Ally decides that the generic functional Roles meet her needs effectively, so she does not need Community-specific functional Roles. With that decision, Ally completes defining her Workflows, Communities, and Roles. She can now proceed to define her publishing requirements.

C H A P T E R 8

Establishing Publishing Requirements

With the design of Content Types, Templates, and user access finished, the modeling and design process is nearly complete. At this point, Ally must establish the publishing requirements for her system.

Noah asks Ally how content is currently served on the site. She tells him that most of their content is served from static HTML files on the Web server, but that content about events is served from a database to dynamically display events based on user queries on parameters such as location, date, and event type. Noah replies that CM System can handle both cases. The Publishing system includes standard Delivery Type configurations for publishing to either a file system or a database. In fact, most customers use the standard Delivery Type configurations included with CM System, updating them with the appropriate local data. Noah notes that the most common reason that a customer would create a new Delivery Type configuration is because they created a custom publishing plugin and need to configure a Delivery Type for it.

Noah also informs Ally that if the Web server is on a remote machine from the CM System server, additional Delivery Types are available for FTP and secure FTP (SFTP). Ally tells him that they intend to host the CM System server on the same machine as the Web server, so she does not need to use FTP.

For the moment, Noah recommends that they focus on file system publishing. Noah reminds Ally that they already decided to use Site Folder publishing when they planned the *Site's Managed Navigation* (see page 92). The Site Folders that Ally sketched out when she planned managed navigation will not only hold Navons and pages that navigation links access, but also other content items that fit into each Folder category. During publishing, CM System creates a set of directories the target Web Server's file system that duplicates the Folders defined for the Site in the navigation pane of Content Explorer. Public Content Items in the Site Folders are published to these directories as HTML pages.

Selecting the Content to Publish

Noah and Ally must first determine how to select the Content Items that will be published. Noah explains that the basic publishing unit in CM System is an Edition. A publishing job is started by initiating the processing of an Edition. An Edition contains a set of Content Lists, which are processed in the order defined in the Edition. The Content Lists select the Content Items that will actually be published. The Editions are associated with Sites, which determine the location where the assembled Content Items will be published.

Noah then outlines the Content List and Edition configurations available in CM

System. When creating a Content List, you must define:

- which Content Items will be published;

  The Content List Generator determines the initial list of Content Items to be published. The initial list is run through an Item Filter to determine the final list.

- which Templates will be used to assemble the selected Content Items;

  The Template Expander determines which Templates will be used for assembly.

- whether all eligible Content Items will be published, or only those that changed since the last publishing run;

  If a Content List is flagged as incremental, then only Content Items that meet one of the following criteria will be processed:

  - the Content Item was Transitioned to a Public State since the last publishing run for the Site; the Content Item will be published to the Site.

  - the Content Item was modified since the last publishing run for the Site; the Content Item will be republished to the Site with the modifications.

  - the Content Item was Transitioned from a Public State to an Archive State since the last publishing run for the Site; the Content Item will be removed from the Site.

  - the Content Item includes a link to a Content Item affected by one of the other criteria; the Content Item will be republished with appropriate modification.

  If the Content List is not flagged as incremental, all Content Items returned from the Item Filter will be published.

- which Delivery Type will be used to deliver the assembled Content Items.

When defining an Edition, you must define:

- which Content Lists will be run;

  The Content Lists determine which Content Items will be published, the Templates that will be used to assemble them, and whether the a full or incremental publish will be processed.

- the Priority of the Edition;

  The priority of an Edition is either Normal or High. An Edition with High Priority will be processed before an Edition with an Normal Priority, and will interrupt the processing of any Normal Priority Editions currently running.

- the unpublishing Behavior of the Edition.

   The options for Behavior are

   - Unpublish then publish

      Edition processing includes unpublishing Content Items that are not in a State specified as either Public or Ignore (typically Quick Edit). HTML pages of Content Items that are not in these States will be removed from the target location. This is the preferred option in most cases.

   - Publish

      Edition processing does not include unpublishing expired Content Items. (Unpublishing may still be available, however, if a Content List explicitly configured to unpublish is associated with the Edition).

Noah explains to Ally that most implementations start with an incremental Edition. An incremental Edition includes one Content List, which is flagged as incremental. The Content List typically uses the sys_SearchGenerator Content List Generator, with a query that selects Content Items from all Content Types. The Item Filter is generally sitefolder. This Item Filter inherits the public Item Filter, which filters out Content Items that are in a State other than Public or Quick Edit (Ignore), but also filters out links to Content Items that are not Public on other Sites managed in CM System. The sys_SiteTemplateExpander is used to publish the Templates associated with the Site. The Edition is typically configured to Unpublish then Publish and with a Normal Priority. Incremental Editions are typically published frequently to keep content on a Site up to date. Based on Noah's recommendation, Ally decides to include an incremental Edition configured as Noah described, with the name *EI_Incremental*.

Noah then describes full Editions. A full Edition publishes all Public Content Items on the Site. While a full Edition can consist of a single Edition, it is more typical to implement at least two different Content Lists: one to select binary Content Items, the other to select non-binary Content Items. The Content Lists use the sys_SiteTemplateExpander, The sitefolder Item filter is used, and the Incremental checkbox is not checked. In the Edition, the binary Content List is typically ordered before the non-binary Content List to ensure that any files targeted by a link is available when the link is generated. The Edition is typically configured to Unpublish then Publish and with a Normal Priority, and is not flagged as Incremental. Full Editions are the first Editions published when an implementation goes live, and are generally published at longer intervals to keep a Site completely up to date. Ally follows Noah's recommendation to include a full Edition in her implementation and names it *EI_Full*.

Many implementations, Noah notes, include a variation on the full Edition that includes only the non-binary Content List. This Edition is generally included to support changes in Site Navigation, such as Folders being moved within the Site. Other than including only the non-binary Content List, the configuration of this Edition is identical to the EI_Full Edition. Ally follows Noah's recommendation and names this Edition *EI_Full_Nonbinary*.

Many customers share Ally's need to publish individual Content Items on demand, Noah says. CM System supports this requirement though demand Editions, often named *Publish Now*. A demand Edition usually includes one Content List, which uses the sys_SelectedItemsGenerator; this Content List Generator selects only the Content Items selected by the user in Content Explorer. The Content List uses the sys_SiteTemplateExpander and is not flagged as incremental. The Edition is configured to Publish (unpublishing processing is not desired in this case) and with a High Priority (allowing it to override any Normal Priority Editions that might be running.

Note that since Ally wants to implement publishing to a file system, the Delivery Type on all of these Content Lists is *filesystem*.

Determining a Publishing Schedule

Once the Editions and Content Lists have been defined, Noah and Ally turn to defining a publishing schedule. Noah notes that, while Editions can be started manually, in a production environment Editions are typically run automatically on a schedule. Common practice, he says, is to publish a Full Edition on some longer schedule, such as weekly or monthly, with incremental Editions being published at shorter intervals, such as daily. Special Demand Editions can be run manually to make high-priority changes to site content.

Noah asks Ally how many pages are currently managed in her Site, and how many new pages are created each day. Ally estimates that the Site generally consists of about 150 HTML pages. Content contributors create about thirty new pieces of content each day, but the overall results is that about ten pages change per day, either being added, modified, or deleted. Noah then asks about the current publishing schedule. Ally replies that the current delivery system is scheduled to publish new pages to the Web site once in the morning, and to remove expired content once in the evening. Once or twice a week, a page must be published or removed before a scheduled run; in that case, the Web Master handles the change manually.

Noah responds that publishing ten pages per day is well within the capabilities of CM System. He suggests that Ally should schedule the incremental Edition to run three times per day to handle routine changes to the Site. Such a frequent publishing schedule should reduce the need to publish individual Content Items on demand. In addition, Noah suggests running the full Edition weekly to ensure that the Site is fully up- to-date, acting as a backup in case the incremental Edition skips a Content Item for some reason. Ally agrees with Noah's recommendations.

NOTE: Publishing schedules are very subjective. Your Publishing schedule should be defined based on the number of Content Items that are routinely created and modified each day in your system. Ally's publishing schedule is appropriate for her needs, but may not be appropriate for your needs. For example, a Web site that included a lot of news might schedule a Normal Edition to run every hour, while a retailer whose stock changes once per season might schedule it to run once or twice per week. The number of pages maintained on a Web site also plays a role in determining how frequently a Full Edition should run. A customer with a large Web site should avoid running a Full Edition each night to avoid performance problems in their server.

Defining Locations and URLs

Noah recommends that during the modeling and design session, they define the directory path to the locations where output files will be published, the file name used for the output files, and the URLs used in links between those files. Noah points out that the paths and URLs will be different in different output locations, or Contexts. The URL of a graphic file used in previewing a Content Item on the CM System server will be different than the URL used in rendering the published page from the production Web server. Each is a different Context.

For each Context, Noah says, they need to define a set of Location Schemes, which generate the actual paths and URLs. A unique Location Scheme can be defined for each combination of Content Type and Template. In most cases, Noah notes, one default Location Scheme is defined that is used by most Content Type/Template combination, with addition Location Schemes for specific combinations that require a different result.

Noah mentions that CM System includes a default Context (and set of Location Schemes) for Preview, so Ally does not have to define how Content Items will be previewed. But she must implement Contexts and Location Schemes for her production environment.

Ally suggests a basic Location Scheme

- The Location Scheme begins with the Folder path to the Content Item.
- The Location Scheme includes a unique identifier for the Content Item.
- The Location Scheme ends with a file extension. For HTML files, the extension would obviously be ".html". For binary files, the extension is the standard extension for the type of binary file (for example, ".gif", ".jpg", or ".pdf").

Noah suggests using the Content Item's id as the unique identifier and shows and a sample location for a Content Item:

```
AboutEnterpriseInvestments/486.jpg
```

Ally does not like the idea of the file name beginning with a number, so she decides to add the string "Item" before the Content Item ID. Noah suggests that the Location Scheme also include the prefix that can be added to assembled Content Items from the Template. For example, a Content Item assembled by a Template that specifies the string "GEN" as the prefix would result in an output location such as:

```
AboutEnterpriseInvestments/GENItem486.jpg
```

He also notes that Content Items other than binaries might have extensions specified. Ally agrees to to change the Location Scheme to include any extension stored for the Content Item before defaulting to ".html".

Ally plans to use this Location Scheme both to define the location where each Content Item will be published and to define the links between Content Items. Noah observes that this plan is often possible, but asks whether Ally's Web server uses a virtual root or the URL is rewritten. In these cases, the link URLs are different than the publishing location and an additional Context and set of Location Schemes is required to generate the correct URLs.

Noah says they have enough information on this subject, and can work out the details during implementation.

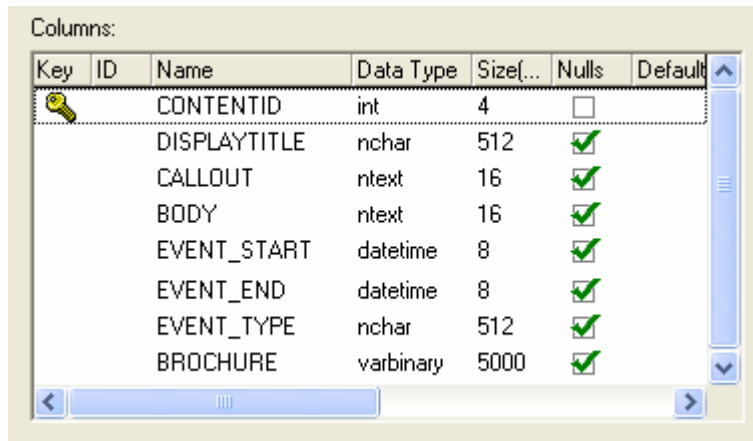# Planning Database Publishing

Ally has already stated that she plans to use CM System's database publishing capabilities to publish Event Content Items to a database. She wants the Site to include pages that allow users to query the database on parameters such as city, date, and event type to dynamically display matching events.

Currently, event content is published once per week, and Ally wants to continue to follow that schedule.

 Although Ally has stated that she plans to publish data from Event Content Items to her database, she has not specified which information she plans to publish. Noah suggests that now she create models of her database publishing table schemas.

Ally's *Event Content Items* (see "Determining Fields for the Event Content Type" on page 52) include a child table editor with any number of event locations and their associated data. She decides to duplicate this structure in her target database tables. In the parent target table, she will omit most metadata fields, which she does not plan to display on the Web Page, and instead include the title, callout (summary) and body fields and the fields directly associated with events. The child table is easy to plan. She includes the Content ID to associate the child table with the parent table, and a seq field to give each child table entry a unique id. Then she includes the fields included in the Content Type's child field set: event_city, event_state, event_address, and event_contact. Users will query on these fields, and the resulting Content Items will be displayed on the Web page along with the parent table fields.
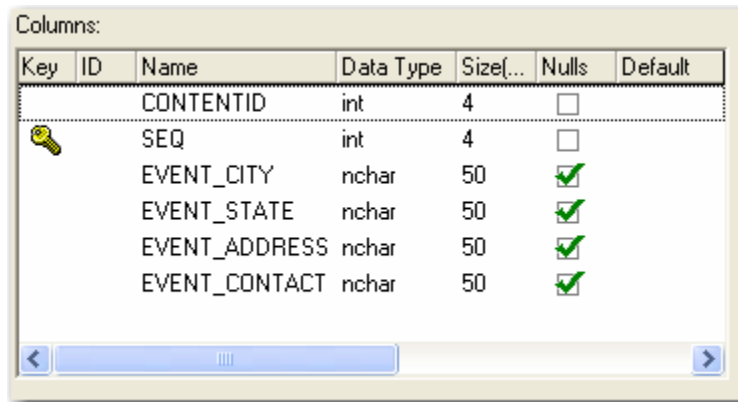
The target tables will resemble:

Figure 64: TARGET_CONTENT table

Figure 65: TARGET_LOCATION table

Ally must give the sketches of these tables, along with the name and port of the database that she wants to publish to the implementers of the system. With this information, they can set up a database publishing Template that will publish to the correct database and enter the correct information into her tables.

Noah explains that database publishing requires minimal publishing configuration. Content Lists are required to select the content to publish, he says. The Content Lists typically use the sys_SearchGenerator and the sys_ListTemplateExpander, specifying the database publishing Templates. The Content Lists must be associated with Editions. The association of a Content List to an Edition requires a Context, but only a very rudimentary Location Scheme is required. Ally wants to implement a simple Location Scheme that names each Content Items *item + content id + .xml*, for example, item455.xml. Noah agrees with her suggestion.

 While in Ally's case, a unique Site for database publishing is not necessary, Noah recommends creating one to make it easier to organize and locate database publishing configurations.

Ally has now completed the major part of the process of modeling and design. Now she and Noah will write her development plan using the specifications they put together during this process. The next section shows illustrates their final development plan.

C H A P T E R   9

Ally's Development Plan

A development plan contains the specifications that the customer and CM System implementer have designed during the modeling and design process. Creating formal specifications not only ensures that the customer and CM System implementer agree about the details of the system being created, but also gives other implementers who are developing the system an exact plan of the components to create.

A development plan describes in detail:

- the system environment
- system fields (these are predefined by CM System)
- shared field sets
- Content Types
- Templates and Slots
- relationships between Content Types, shared field sets, Templates and Slots
- Global Template and Managed Navigation requirements
- Publishing requirements
- Integration with other systems, if any
- Workflows
- Communities
- Roles
- special components such as extensions

Ally's development plan is the subset of the FastForward development plan that covers what has been modeled and designed in this chapter. Its details include modifications to the FastForward plan that Ally and Noah have made during modeling and design.

Note: Since an official development plan (also called a Statement of Work) is also a legal agreement between the two parties, there are portions of the plan that we will not include in this section, including goals, inclusions, and exclusions.

In an actual plan, the discussion of the system's architecture would probably be more detailed, and cover details of security, authentication, and email used for Notifications.

On the last page of an actual plan, both parties would be required to sign off before any development began.

System Architecture

CM System will be installed in development, testing, and production environments. In each of these environments, the CM System Server will be hosted on a Windows XP server machine, and the database will be installed to a SQL Server 2000 instance.

Implementers will create the original system on the development server and then deploy its components to the testing server. If testing demonstrates that functionality is not working as intended, implementers may send components back to the development server for additional work. Once testing confirms that the system, including publishing, is working correctly on the testing server, implementers will deploy its components to the production server, so that content contributors can begin creating content and content can be published to the production Web Server and database.
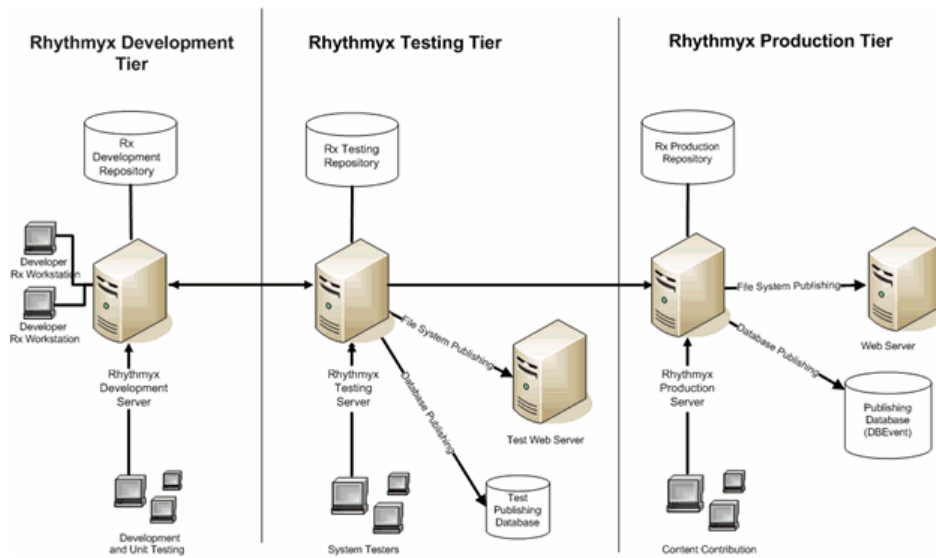
Figure 66: Ally's CM System
System

System Fields

Name	Label	Control Name	Data Type	Rules	Format (size)
sys_communityid	Community:	sys_DropDownSingle	integer	Required, Hidden	none
sys_contentexpirydate	Expiration Date:	sys_CalendarSimple	datetime		none
sys_contentstartdate	Start Date:	sys_CalendarSimple	datetime	Required	none
sys_currentview		sys_HiddenInput		Hidden	none
sys_hibernateVersion		sys_HiddenInput	integer	Hidden	none
sys_lang	Locale:	sys_DropDownSingle	text	Required, Hidden	16
sys_pathname	Path Name:	sys_EditBox	text		255
sys_pubdate	Pub Date:	sys_CalendarSimple	datetime		none
sys_reminderdate	Reminder Date:	sys_CalendarSimple	datetime		none
sys_suffix	Suffix:	sys_EditBox	text	Hidden	50
sys_title	System Title:	sys_EditBox	text	Required	255
sys_workflowid	Workflow:	sys_DropDownSingle	integer	Required, Hidden	none

Field Descriptions:

Field	Description
sys_communityid	The id for the community in which the content item was created.
sys_contentexpirydate	May be used to trigger aging transitions out of the public workflow state.
sys_contentstartdate	May be used to trigger aging transitions into the public workflow state.
sys_currentview	(Required for internal system operation)
sys_hibernateVersion	(Required for internal system operation)
sys_lang	The id for the locale in which the content item was created (for multi-lingual support)..
sys_pathname	(Deprecated) Used for creation of delivery location path in publishing; This has been replaced by site folder publishing functionality.

sys_pubdate	May be used to trigger aging transitions for publishing.
sys_reminderdate	May be used to trigger aging transitions for "reminder" notifications.
sys_suffix	Used in publishing delivery location schemes for filename generation.
sys_title	The internal title of the content item, which will be displayed in the Content Explorer.
sys_workflowid	The id for the workflow in which the content item was created.

Default Values:

Field	Value
sys_contentstartdate	<Today's date> (sys_DateFormat ext., with yyyy-MM-dd)
sys_suffix	.html
sys_communityid	PSXUserContext/User/SessionObject/sys_community
sys_lang	PSXUserContext/User/SessionObject/sys_lang

Shared Field Sets

Shared

Name	Label	Rules	Description	Control	Data Type/ Storage Size
displaytitle	Title	Required	The title shown to users.	sys_EditBox	text/512
body	Body		The main body of content. Since the sys_EditLive control is used, the body is stored in rich text format and may include inline links and images.	sys_EditLive	text/max
filename	File name	Hidden	The file name of the item when it is published.	sys_EditBox	text/512
keywords	Keywords		Search terms that are not part of the item's content and are inserted into markup tags.	sys_TextArea	text/1024
callout	Callout		A synopsis of the body content.	sys_EditLive	text/max
description	Description		Search phrases that are not part of the item's content and are inserted into markup tags.	sys_TextArea	text/1024
webdavowner	WebDAV Owner	Hidden	Stores the user with a lock on the Content Item when content is uploaded through WebDAV.	sys_TextArea	text/255

SharedImage

Name	Label	Rules	Description	Control	Data Type/ Storage Size
img1	Image	Required	Field that uploads the image.	sys_file	binary/max

Name	Label	Rules	Description	Control	Data Type/ Storage Size
img1_filename	Image Filename	Required Read only	File name of the uploaded image. sys_FileInfo extracts this data for system processing or user interface display. sys_webImageFX requires this data to store and display the file.	sys_EditBox	text/10
img1_size	Image File Size	Hidden	File size of the uploaded image. sys_FileInfo extracts this data for system processing or user interface display.	sys_EditBox	integer/none
img1_type	Image Mime Type	Required, Read only	MIME type of the uploaded image. sys_FileInfo extracts this data which sys_File requires to display the file in a browser.	sys_EditBox	text/256
img1_ext	Image Extension	Hidden, Required	Extension of the uploaded image. sys_FileInfo extracts this data which is required to display the file in a browser.	sys_EditBox	text/10
img1_height	Image Height		Height of the uploaded image. The sys_imageInfoExtractor pre-exit extracts this data for system processing or user interface display.	sys_EditBox	integer/none
img1_width	Image Width		Width of the uploaded image. The sys_imageInfoExtractor pre-exit extracts this data for system processing or user interface display.	sys_EditBox	integer/none
img_alt	Image Alt Text		Alternate text shown on screen if image does not render.	sys_EditBox	text/512
img2	Mini	Hidden	Field that uploads thumbnail of image.	sys_file	binary/max

Name	Label	Rules	Description	Control	Data Type/ Storage Size
img2_filename	Mini Filename	Hidden	File name of the uploaded thumbnail image. sys_FileInfo extracts this data for system processing or user interface display. sys_webImageFX requires this field to display the file.	sys_Editbox	text/512
img2_size	Mini File Size	Hidden	File size of the uploaded thumbnail image. sys_FileInfo extracts this data for system processing or user interface display.	sys_Editbox	integer/none
img2_type	Mini Mime Type	Hidden	MIME type of the uploaded thumbnail image. sys_FileInfo extracts this data which is required to display the file in a browser.	sys_Editbox	text/256
img2_ext	Mini Extension	Hidden	Extension of the uploaded thumbnail image. sys_FileInfo extracts this data which sys_File requires to display the file in a browser.	sys_Editbox	text/10
img2_height	Mini Height	Hidden	Height of the uploaded thumbnail image. The sys_imageInfoExtractor pre-exit extracts this data for system processing or user interface display.	sys_EditBox	integer/none
img2_width	Mini Width	Hidden	Width of the uploaded thumbnail image. The sys_imageInfoExtractor pre-exit extracts this data for system processing or user interface display.	sys_EditBox	integer/none

Slots and Templates

Note: The following Slot and Template descriptions are based on those Ally and Noah have discussed during Modeling and Design. Therefore, only a portion of Slots and Templates associated with the Generic, Image, and Event Content Types are listed. All of these Slots and Templates (except EventDB) exist in FastForward; some of them have been specifically mentioned in the Modeling and Design sections; others have not been discussed but are included because they are associated with one another.

Slots:

Name	Allowed Content Types	Description
Sidebar Slot	all	General sidebar for related content.
List Slot	all	Lists of links to other content.
sys_inline_link	all	System Slot for inline links.

Shared Templates:

Name	Description	Slots	Type	Publish
D - EI Generic	*Dispatch Template for Generic pages. Selects either the P - EI Generic Template or the P - EI Generic Category Template.* (see page 38)	None	Page	Default
P - EI Generic	Generic full page, renders title, body, and callout (summary).	Sidebar, List	Page	Never (because Dispatch Template will specify when it is published)
P - EI Generic Category	Generic full page for items that hold product categories. Renders title, body and callout (summary)	Sidebar, List	Page	Never (because Dispatch Template will specify when it is published)
S - Title Link	Rendering of title field as a link to the full page.	None	Snippet	Never
S - Title Callout Link	Renders a Title and Callout Link.	None	Snippet	Never
S - Callout	Renders the shared callout field.	None	Snippet	Never

Shared Template Screenshots:

P-EIGenericPage
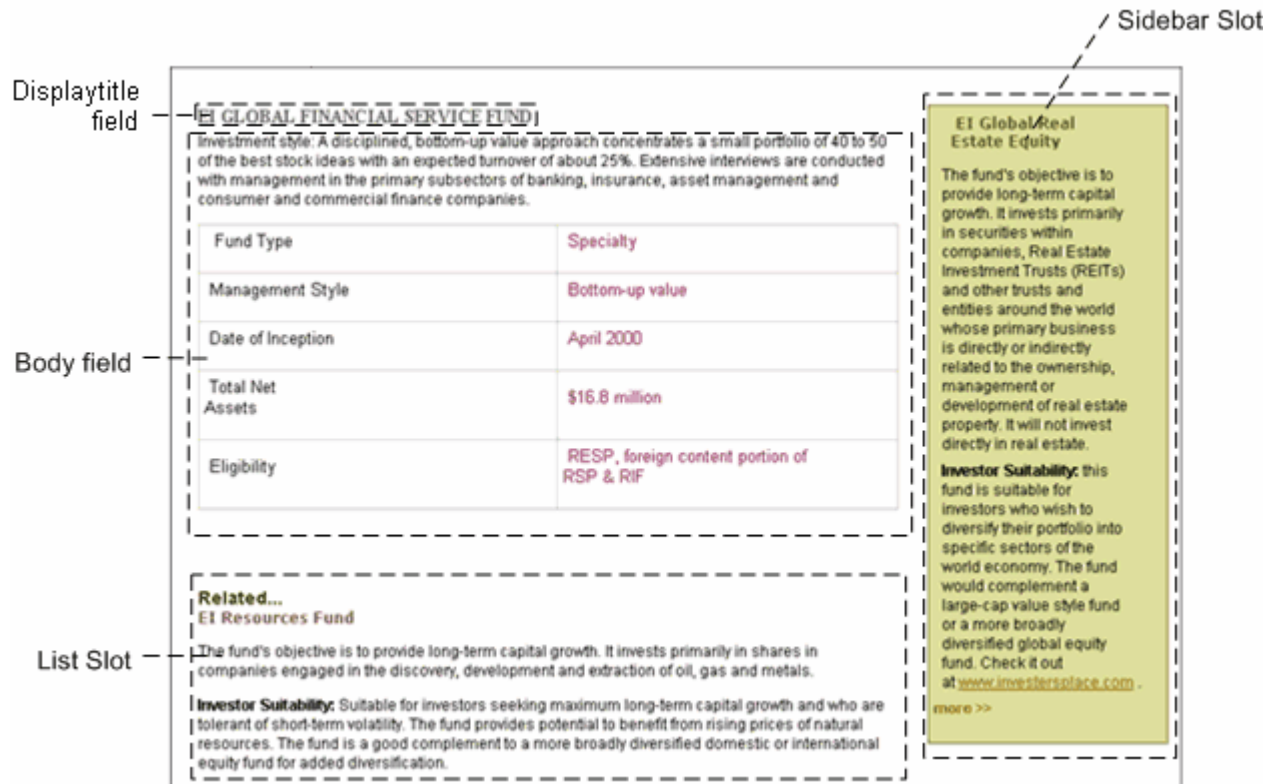
Figure 67: Generic Content Type Template

P-EIGenericCategoryPage
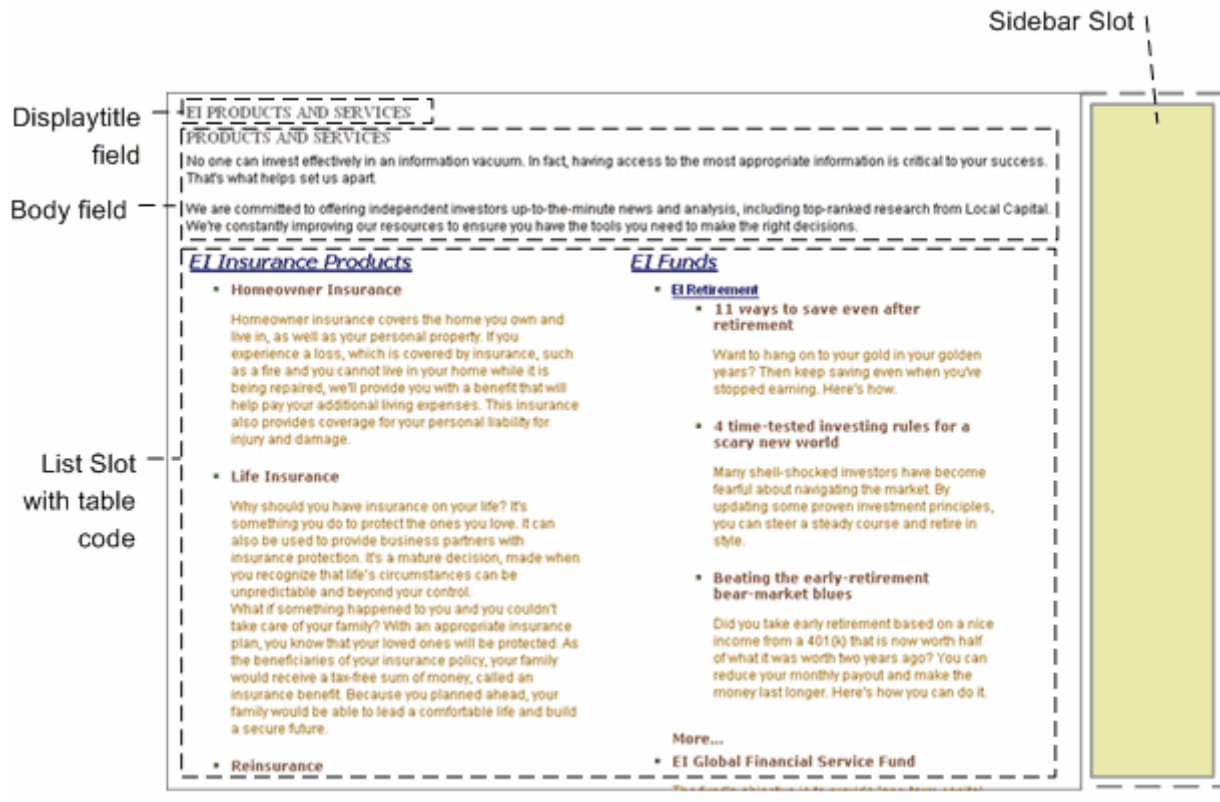
Figure 68: Generic Template for Major Products

S-TitleLink
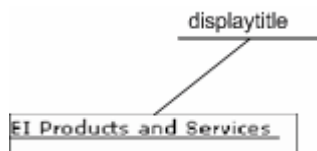
Figure 69: S - Title Link Template

S-TitleCalloutLink
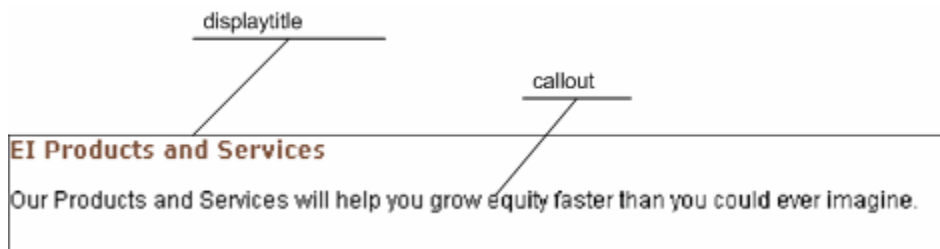
Figure 70: S - Title Callout Link Template

<u>S-Callout</u>
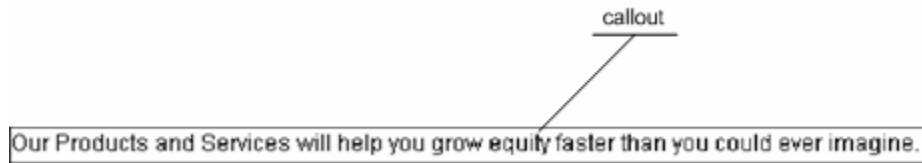
Figure 71: S - Callout Template

Type-Specific Templates:

Name	Description	Slots	Type	Publish
B-Image	Returns the binary stream from the image shared group img1 field, setting the mime type as specified by the Image mime type field. This is never inserted into a Slot; Image Snippets use it to acquire the image.	None	Image	Always
S-Flash	Displays streaming images,	None	Image	Never
S-Image	Displays just the image as a snippet	None	Image	Never
S - Image and Title	same as S - Image, but with the display title centered underneath the image.	None	Image	Never
S - Title, Callout and More	Renders an Image, Title, callout and more Link	Image Link	Generic	Never
P - Event Query (to be created during implementation)	Enables users to enter information about Events and pull up lists of matching events from the database.	to be determined	to be determined	Always
P - EI Event	EI Full Page Event	List, Sidebar	Event	Default

Type-Specific Template Screenshots

(B - Image, S - Flash, and P - Event Query are not applicable for screenshots)

S-Image
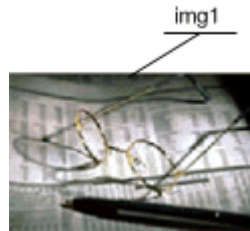
Figure 72: S - Image Template

S-ImageandTitle
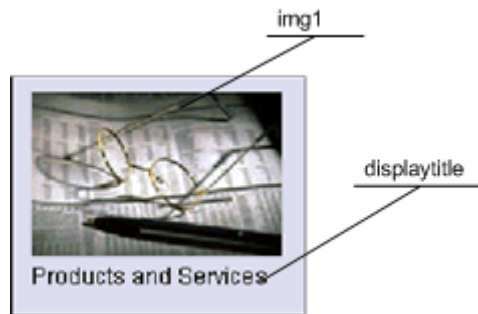
Figure 73: S - Image and Title Template

S-TitleCalloutAndMore
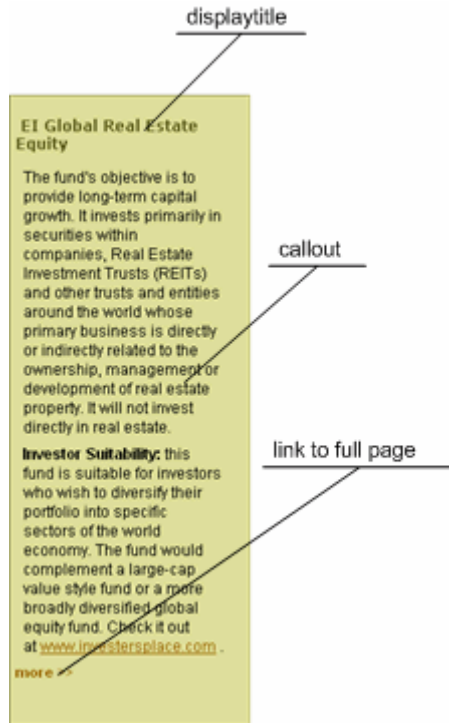
Figure 74: S - Title Callout and More Template

P-EIEvent

Figure 75: P - EI Event

Database Publishing Template:

Name	Description	Slots	Type	Publish
EventDB	Template for storing Event Content Type information in a database.	None	Event	Default

TargetDatabaseTables:

	Column Name	Data Type	Length	Allow Nulls
🔑	CONTENTID	int	4	
	DISPLAYTITLE	nchar	512	✓
	CALLOUT	ntext	16	✓
	BODY	ntext	16	✓
	EVENT_START	datetime	8	✓
	EVENT_END	datetime	8	✓
	EVENT_TYPE	nchar	512	✓

Figure 76: TARGET_CONTENT table

Columns:

Key	ID	Name	Data Type	Size(...	Nulls	Default
		CONTENTID	int	4	☐	
🔑		SEQ	int	4	☐	
		EVENT_CITY	nchar	50	☑	
		EVENT_STATE	nchar	50	☑	
		EVENT_ADDRESS	nchar	50	☑	
		EVENT_CONTACT	nchar	50	☑	

Figure 77: TARGET_LOCATION table

Content Types

Generic

Field	Field Name	Source	Type (Size)	Control	Rules
System Title	sys_title	system	text (255)	sys_EditBox	Required
Title	displaytitle	shared	text (512)	sys_EditBox	Required
Start Date	sys_contentstartdate	system	datetime	sys_calendarsimple	Required
Expiration Date	sys_contentexpirydate	system	datetime	sys_calendarsimple	Optional
Reminder Date	sys_reminderdate	system	datetime	sys_calendarsimple	Optional
Keywords	keywords	shared	text (1024)	sys_textarea	Optional
Description	description	shared	text (1024)	sys_textarea	Optional
Callout	callout	shared	text (max)	sys_textarea	Optional
Body	body	shared	text (max)	sys_EditLive	Optional
File Name	filename	shared	text (512)	sys_EditBox	Hidden, Optional
Suffix	sys_suffix	system	text (50)	sys_EditBox	Hidden, Optional
Community	sys_communityid	system	integer	sys_DropDownSingle	Hidden, Required
Workflow	sys_workflowid	system	integer	sys_DropDownSingle	Hidden, Required
Locale	sys_lang	system	text (50)	sys_DropDownSingle	Hidden, Required
	sys_currentview	system	text	sys_HiddenInput	Hidden, Optional
WebDAV Owner	webdavowner	shared	text (255)	sys_TextArea	Hidden, Optional
	sys_hibernateVersion	system	integer	sys_HiddenInput	Hidden, Optional
Usage	usage	local	text(1)	sys_DropDownSingle	Required

Image

Field	Field Name	Source	Type (Size)	Control	Rules
System Title	sys_title	system	Text (255)	sys_EditBox	Required
Title	displaytitle	shared	Text (512)	sys_EditBox	Required
Start Date	sys_contentstartdate	system	datetime	sys_CalendarSimple	Required

Field	Field Name	Source	Type (Size)	Control	Rules
Expiration Date	sys_contentexpirydate	system	datetime	sys_CalendarSimple	Optional
Reminder Date	sys_reminderdate	system	datetime	sys_CalendarSimple	Optional
Description	description	shared	text (1024)	sys_TextArea	Optional
Image	img1	shared	binary (max)	sys_File	Required
Image file name	img1_filename	shared	text (512)	sys_EditBox	Optional
Image Extension	img1_ext	shared	text (50)	sys_EditBox	Required
Image Mime Type	img1_type	shared	text (256)	sys_EditBox	Optional, Read-only
Image Height	img1_height	shared	integer	sys_EditBox	Optional
Image Width	img1_width	shared	integer	sys_EditBox	Optional
Image Alt Text	img_alt	shared	text (512)	sys_EditBox	Optional
Image Category	img_category	local	text(50)	sys_DropDownSingle	Optional
Image file size	img1_size	shared	integer	sys_EditBox	Optional
File Name	filename	shared	text (512)	sys_EditBox	Hidden, Optional
Suffix	sys_suffix	system	text (50)	sys_EditBox	Hidden, Optional
Mini	img2	shared	binary (max)	sys_File	Hidden, Optional
Mini File Name	img2_filename	shared	text (512)	sys_EditBox	Hidden, Optional
Mini Extension	img2_ext	shared	text (50)	sys_EditBox	Hidden, Optional
Mini Mime Type	img2_type	shared	text (256)	sys_EditBox	Hidden, Optional
Mini Height	img2_height	shared	integer	sys_EditBox	Hidden, Optional
Mini Width	img2_width	shared	integer	sys_EditBox	Hidden, Optional
Mini File Size	img2_size	shared	integer	sys_EditBox	Hidden, Optional

Field	Field Name	Source	Type (Size)	Control	Rules
Community	sys_communityid	system	integer	sys_DropDownSingle	Hidden, Required
Workflow	sys_workflowid	system	integer	sys_DropDownSingle	Hidden, Optional
Locale	sys_lang	system	text (16)	sys_DropDownSingle	Hidden, Optional
	sys_currentview	system	text	sys_HiddenInput	Hidden, Optional
WebDAV Owner	webdavowner	shared	text (255)	sys_HiddenInput	Hidden, Optional

Event

Field	Field Name	Source	Type (Size)	Control	Rules
System Title	sys_title	system	text (255)	sys_EditBox	Required
Title	displaytitle	shared	text (512)	sys_EditBox	Required
Start Date	sys_contentstartdate	system	datetime	sys_calendarsimple	Required
Expiration Date	sys_contentexpirydate	system	datetime	sys_calendarsimple	Optional
Reminder Date	sys_reminderdate	system	datetime	sys_calendarsimple	Optional
Keywords	keywords	shared	text (1024)	sys_textarea	Optional
Description	description	shared	text (1024)	sys_textarea	Optional
Callout	callout	shared	text (max)	sys_textarea	Optional
Body	body	shared	text (max)	sys_EditLive	Optional
Event Start Date	event_start	local	datetime	sys_calendarsimple	Optional
Event End Date	event_end	local	datetime	sys_calendarsimple	Optional
Event Location	event_location	local		sys_table	Required
Event Type	event_type	local	text (255)	sys_DropDownSingle	Optional
File Name	filename	shared	text (512)	sys_EditBox	Hidden, Optional
Suffix	sys_suffix	system	text (50)	sys_EditBox	Hidden, Optional
Community	sys_communityid	system	integer	sys_DropDownSingle	Hidden, Required
Workflow	sys_workflowid	system	integer	sys_DropDownSingle	Hidden, Required

Field	Field Name	Source	Type (Size)	Control	Rules
Locale	sys_lang	system	text (50)	sys_DropDownSingle	Hidden, Required
	sys_currentview	system	text	sys_HiddenInput	Hidden, Optional
WebDAV Owner	webdavowner	shared	text (255)	sys_TextArea	Hidden, Optional
	sys_hibernateVersion	system	integer	sys_HiddenInput	Hidden, Optional

Event Location Child Set

Field	Field Name	Type (Size)	Control	Rules
Event City	event_city	text (50)	sys_editBox	Optional, Show in Summary
Event State	event_state	text (50)	sys_editBox	Optional, Show in Summary
Event Address	event_address	text (255)	sys_textArea	Optional
Event Contact	event_contact	text (255)	sys_textArea	Optional

Global Template

The Global Template will have a typical "C-clamp" structure, and include CM System Navigation Content Types in Slots to compose a top navigation bar, left navigation links, bottom navigation links, and breadcrumbs. The top banner of the Global Template will include a static logo image, a search control, and a drop list for selecting a region or country.
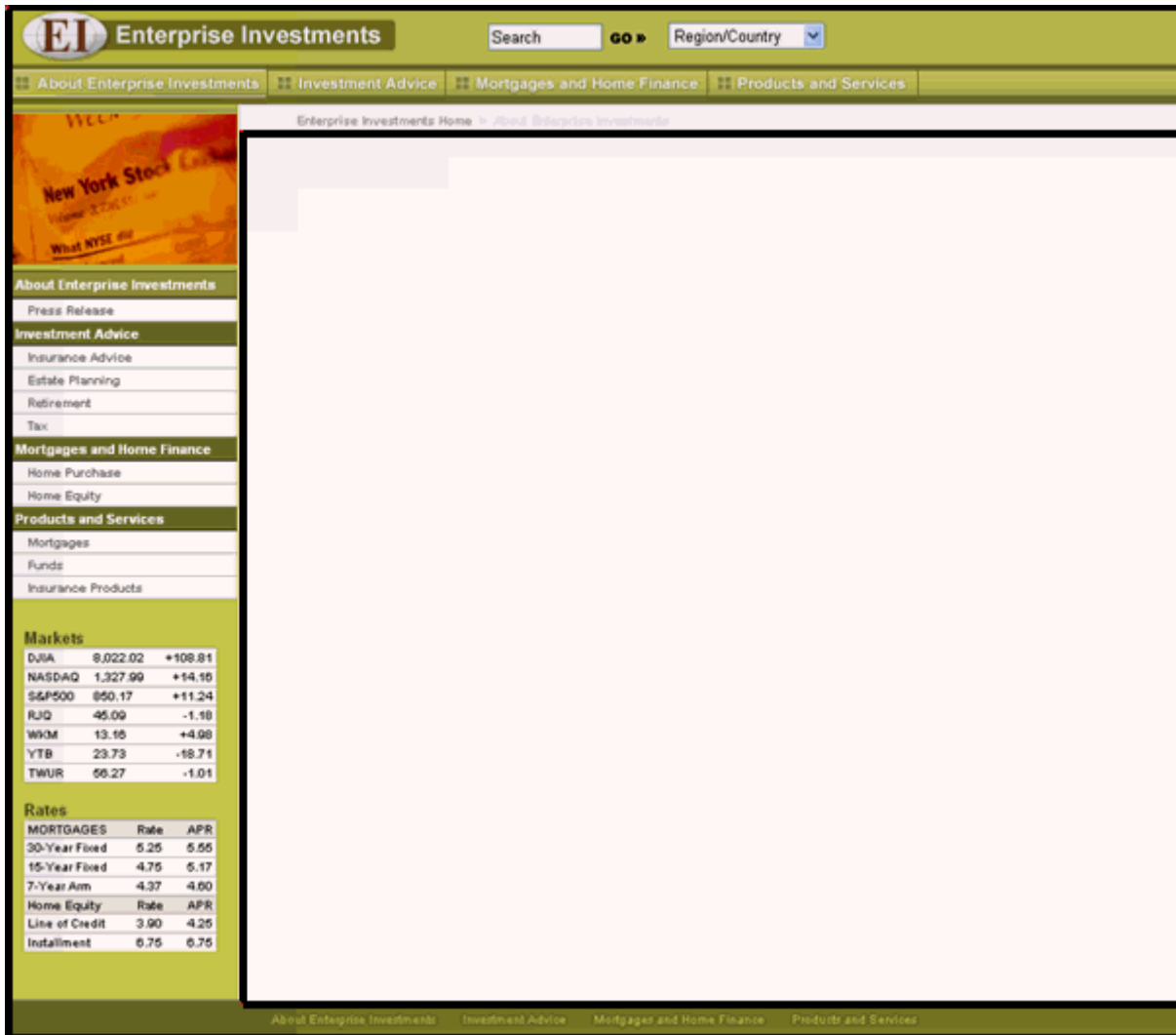
Figure 78: Global Template

Managed Navigation

Managed Navigation uses the Navon, NavTree, and NavImage Content Types that are installed with CM System's FastForward.

Managed Navigation is implemented through use of the Site Folder structure in CM System, which will contain the appropriate navigational Content Types. The Site Folder structure also corresponds to the publishing locations on the Site. Detailed information regarding Managed Navigation concepts is included in additional documentation.

The Site structure in Content Explorer (and on the published Site) will include the following folders:
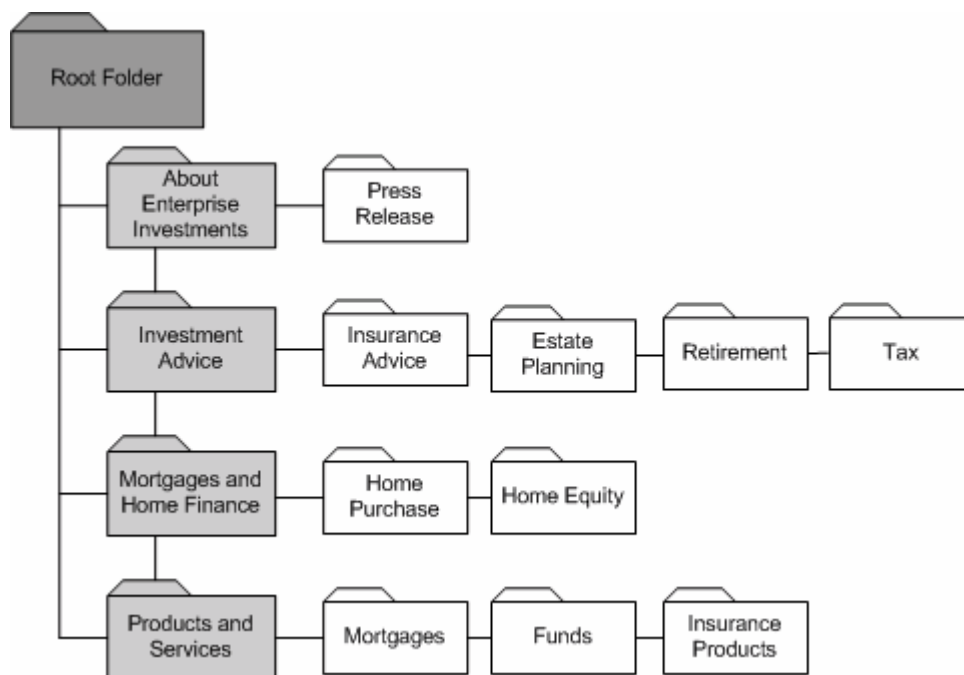
Figure 79: Site Folders for Managed Navigation

It will also include other folders that store Content Items that do not appear in navigation elements of the Site, such as an Images folder for storing images that appear on the Site.

Publishing

Site Folder

The Web Site's structure will be controlled using the Site Folder functionality in CM System. The plan for the initial Site Folder Structure is the following, although additional folders may be added during implementation. The Files and Images folders will not contain navigation components, but will store files and images for use on Web pages.
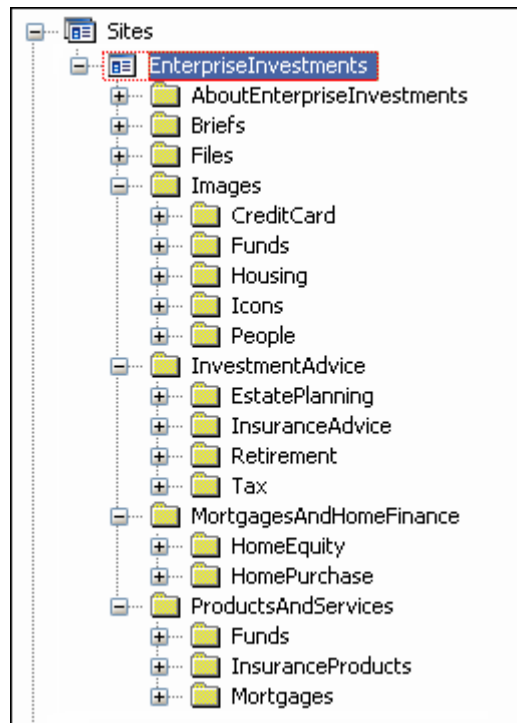
Figure 80: Initial Site Folder Structure

Delivery Type

No custom Delivery Types are required. The standard filesystem and database Delivery Types will be used.

Sites

Two Sites will be defined, one for file system publishing and the other for database publishing. Both will use the same Site Folder root.

File system Site:

Name	Value
Site Name	Enterprise_Investments

Description	Represents the Enterprise Investments web site
Rhythmyx Path	//Sites/EnterpriseInvestments
Global Template	EI Global Template
Published URL	http://127.0.0.1:9911/EI_Home
Published Path	../EI_Home.war

Database publishing Site

Name	Value
Site Name	Enterprise_Investments_DB
Description	Database publishing Site for Enterprise Investments
Rhythmyx Path	N/A
Global Template	N/A
Published URL	N/A
Published Path	N/A

Content Lists

For each site, the following Content Lists will be defined:

Description	File system name	Database Name
Incremental publish	rffEiIncremental	rffEiIncremental_DB
Full publish, binary only	rffEiFullBinary	rffEiFullBinary_DB
Full publish, nonbinary only	rffEiFullNonbinary	rffEiFullNonbinary_DB
Demand publish (Publish Now	rffEiPubilshNow	rffEiPubilshNow_DB

Editions

Name	Content Lists	Incremental?	Priority	Behavior	Description
EI_Incremental	rffEiIncremental	Yes	Normal	Unpulbish then publish	Publishes new and modified Content Items in the specified Site Folders
EI_Full	rffEiFullBinary rffEiFullNonbinary	No	Normal	Unpublish, then Publish	Publishes all currently Public Content Items in the specified Site
EI_Full_Nonbinary	rffEiFullNonbinary	No	Normal	Unpublish then Publish	Publishes all currently Public nonbinary Content Items in the specified Site.
EI_Publish_Now	rffEiPublishNow	No	High	Publish	Publishes the Content Items currently selected by the user.
EI_Incremental_DB	rffEiIncremental_DB	Yes			

Content Lists

The initial Content Lists that will support Ally's publishing system are the following:

- Full Binary - includes all binary Content Items that are ready to be published
- Full Non-binary - includes all non-binary Content Items that are ready to be published
- Incremental - includes all new and modified Content Items that are ready to be published
- Unpublish - includes all published Content Items that are ready to be unpublished (archived)
- Event Items to Database - includes all Event Content Items that are ready to be published

Location Schemes (Delivery Locations)

Purpose	Location Scheme	Example
location to publish Content Items during file system publishing	site folder path +Template prefix + "item" + sys_contentid + (stored suffix or .html)	AboutEnterpriseInvestments/GENitem486.jpg
location that published content items will use in links for accessing other content items	/EI_Home + site folder path +Template prefix + "item" + sys_contentid + (stored suffix or .html)	/EI_Home/AboutEnterpriseInvestments/GENitem486.jpg
representation of what has been published to database for use in the database publishing log	item + content id + .xml	item455.xml

Additional Database Publishing Details

The purpose of publishing Event Content Items to a database is to include pages on the Site that allow users to query the database on parameters such as city, date, and event type to dynamically display matching events. Users can link on matching events to see the *P - EI Event* (see page 124) page for the event. The Template for the page that queries the Event target database will be created during implementation.

The target database resides on the same MSSQL Server as the CM System database, EIServer, and is named targetdb. Content is published to it over port 1433.

Workflow

Standard Workflow:

Names listed under each transition are Roles that will be assignees of the "from" State of the Transition.
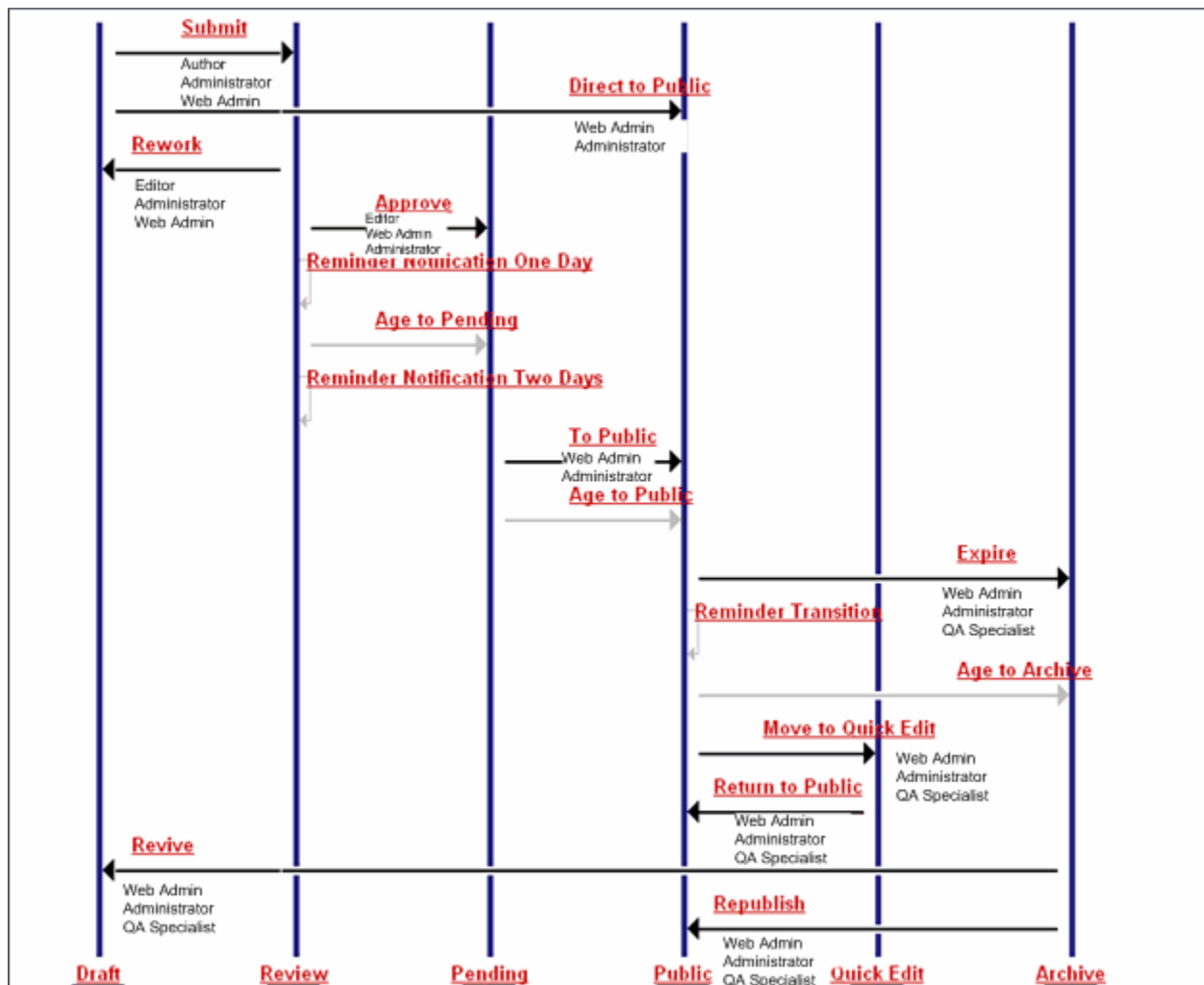
Figure 81: Ally's Final Workflow

Reader Roles:

- in Draft State - Editor Role
- in Review State - Author and QA Roles
- in Pending State - Author, Editor, and QA Roles
- in Publish State - Author, Editor, and RxPublisher Roles
- in Quick Edit State - Author and Editor Roles
- in Archive State - Author Role

Aging Details:

- an Age to Public Transition occurs after the Content Item's Start Date has passed
- an Age to Archive Transition occurs after a Content Item's End Date has passed.
- an Age to Pending Transition occurs after 3 days.

Notification Details:

- a reminder notification is sent to Roles in the Public State after the Content Item's Reminder Date has passed.
- a reminder notification is sent to Roles in the Review State telling them that an item will automatically move to the Pending State in three days if they do not act on it. (sent first day item is in State)
- a reminder notification is sent to Roles in the Review State telling them that an item will automatically move to the Pending State in two days if they do not act on it (sent second day item is in State)
- a reminder notification is sent to Roles in the Review State telling them that an item will automatically move to the Pending State in two days if they do not act on it. (sent third day item is in State)
- a notification is sent to Roles in the Archive State after an Age to Archive transition.
- a notification is sent to Roles in the Archive State after an Expire transition
- a notification is sent to Roles in the Draft State after a Rework transition. The Notification will include the Editor's comment.

Ad hoc Assignees:

- Author Role in Draft State

Required Comments:

- Rework Transition. The comment will be included in the Notification sent.

Required Approvals:

- Approve Transition - Web Admin and Editor are both required approvers.

Simple Workflow:

Names listed under each transition are Roles that will be assignees of the "from" State of the Transition.
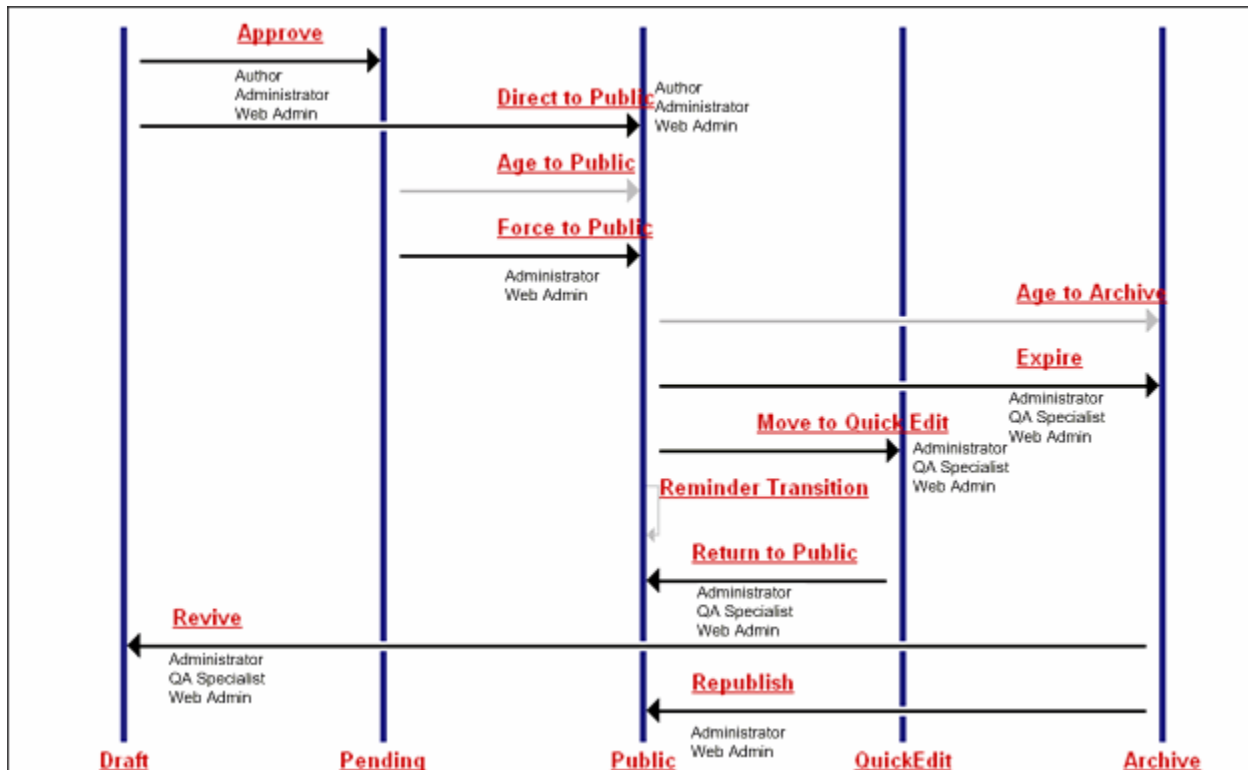
Figure 82: Ally's Simple Workflow

Reader Roles:

- in Draft State - Editor Role
- in Pending State - Author, Editor, and QA Roles
- in Public State - Author, Editor, and RxPublisher Roles
- in Quick Edit State - Author and Editor Roles
- in Archive State - Author Role

Aging Details:

- an Age to Public Transition occurs after the Content Item's Start Date has passed
- an Age to Archive Transition occurs after a Content Item's End Date has passed.

Notification Details:

- a reminder notification is sent to Roles in the Public State after the Content Item's Reminder Date has passed.
- a notification is sent to Roles in the Archive State after an Age to Archive transition.
- a notification is sent to Roles in the Archive State after an Expire transition

Communities and Roles

Communities and Community Roles

The following Communities will be configured initially, although the implementation process may reveal that more Communities are needed to support the management of the Enterprise Investments Site.

- Enterprise Investments
- Enterprise Investments Admin

Each Community will have a corresponding Community Role associated with it; EI_Members for Enterprise Investments and EI_Admin_Members for Enterprise Investments Admin.

Workflow Roles

The initial Workflow Roles consist of those used in the Standard and Simple Workflows:

- Author
- Administrator
- Web Admin
- Editor
- QA Specialist
- RxPublisher

C H A P T E R   1 0

Next Steps

Once you have drafted your development plan, you have completed the modeling and design process. While it is likely that you will revise the plan during implementation of your system, the modeling and design process is essentially over at this point.

You can now begin implementation. For details about implementation, see the *CM System Implementation Guide*.

Index