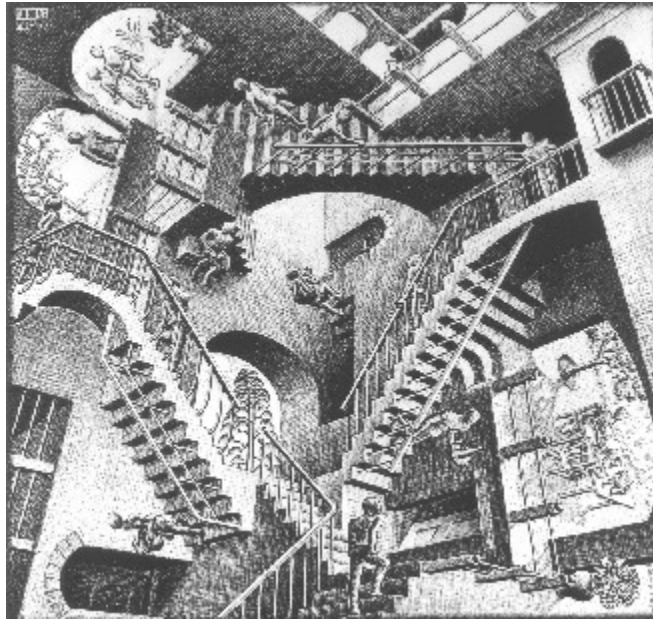


Memory Palaces, Escher, And XML

We face an awe-inspiring prospect in the 21st century—the unification of the fund of knowledge for the entire human species. This article re-examines the medieval Memory Palace concept, a powerful mnemonic device, and the surrealistic art of M. C. Escher to uncover design patterns for information architecture that will help us build XML-based tools for navigating very large information spaces.



[Relativity, drawing by M.C. Escher, Copyright © by Cordon Art. All Rights Reserved.](#)

Dale Hunscher
South Wind Design, Inc.
Ann Arbor, Michigan
(734) 821-0036

Leonard S. Rann
Micromedex, A Thomson Healthcare Company
Englewood, Colorado
(303) 486-9544

Confidential—Proprietary Materials

Last Updated 10/10/2000 3:47 PM

Materials copyright © 2000 by Dale Hunscher and Leonard Rann. All Rights Reserved.

Table Of Contents

| | |
|--------------------------------------------------------------------------|----|
| <i>Introduction</i> | 1 |
| <i>Memory Palaces</i> | 3 |
| History Of The Memory Palace | 3 |
| How Memory Palaces Work | 3 |
| Benefits Of The Memory Palace Practice | 4 |
| Modern Memory Palaces: The Age Of The Library | 4 |
| Post-Modern Memory Palaces: The "Small e" Paradigm Shift | 4 |
| On The Threshold Of Unification | 5 |
| The Current Limits Are Technological And Political | 6 |
| <i>M. C. Escher: Questioning The Assumptions Behind Perception</i> | 7 |
| The Link Between Escher's Art And Knowledge Representation | 7 |
| Choosing And Changing Points Of Reference | 8 |
| <i>Design Patterns For Information Architecture</i> | 9 |
| Providing Support For Information Space Navigation | 9 |
| Separating Knowledge And Content | 10 |
| Design Patterns For Information Architecture | 10 |
| <i>Thomson Healthcare's Challenge</i> | 12 |
| Background | 12 |
| The Problem | 12 |
| The Solution | 12 |
| Some requirements of an knowledge object based system | 13 |
| Architectural Components of the Micromedex Repository | 14 |
| The Implementation | 14 |
| <i>XML Approaches To Memory Palace Construction</i> | 23 |
| Possible Approaches | 23 |
| XTM | 23 |
| RDF | 23 |
| <i>Conclusions</i> | 24 |
| <i>About The Authors</i> | 25 |
| <i>References And Recommended Readings</i> | 26 |
| Memory Palaces | 26 |

Knowledge Representation And Information Architecture 26
M. C. Escher 26
XML-Related Design Patterns 26
Information Architecture 26

Introduction

We face an awe-inspiring prospect in the 21st century—the unification of the fund of knowledge for the entire human species. The amount of information accessible to the average person via the Internet is already vast, and growing in volume exponentially as more and more repositories are brought online. A sophisticated organizational or governmental user with access to private databases is in a position to reach an even larger volume of information.

Unlike physical space, which can be navigated (for any practical purpose) only in accordance with the laws of Newtonian physics, information spaces permit instantaneous navigation—if only we know where we want to go and how to get there! This turns out to be a more difficult problem than we, the denizens of the 20th and now the 21st century, could have imagined before it was possible. The problem in information space navigation is that less is often more; the less information we have access to, the more likely we are to find what we seek. The reason for this is quite simple: we are often looking for something we cannot yet adequately describe. The "do what I mean" problem of systems analysis and design turns into the "find what I mean" problem in the world of information architecture.

Full-text search can at least provide some start at selecting relevant information objects, but if we do not know what search terms are most likely to bring relevant information to the forefront, the truly relevant documents may not bubble to the top of what is often a huge result set. So we are faced with a seemingly unprecedented problem as we approach even vaster information spaces.

Memory Palaces

Fortunately, it turns out the problem is not unprecedented after all. Even in classical Greece and Rome, the problem of memory was on people's minds. In the days before the invention of the printing press, tangible and durable forms of information storage were unimaginably expensive. Only the most important information could be recorded, because the bandwidth of the available recording techniques was very small. Whether the result is handwritten hieroglyphics on papyrus or stone engravings, the time and effort and human knowledge required to record the information was close to priceless. It is not surprising we have so little written material from those times—in part because of the destructive power of entropy in the centuries that have elapsed, but in the main because so little was recorded in the first place.

It turns out there were other ways of recording information in those times, more ephemeral but at the same time strikingly similar to the means we now have at our disposal. Writers of classical Greece and Rome spoke both of natural memory and artificial memory. A powerful metaphor appears in those essays on memory, a metaphor that points to the key concept underlying artificial memory: the Memory Palace.

A Memory Palace was an imaginary architectural space, a building or complex of buildings constructed in the imagination of a practitioner of the art of memory, in which concepts were consciously and painstakingly stored. The stored concepts—sentences in a

speech, verses in an epic poem, whatever—could then be accessed sequentially by traversing the imaginary space as if walking through it in real life. Because the space was imaginary and the laws of physics did not apply, one could also jump at the speed of thought from any point to any other in the Memory Palace.

In the modern world, the Memory Palace need no longer exist only in the mind of an individual, and so need not be limited to a single lifespan or the fund of knowledge of any one person. Our Memory Palaces can be stored on internetworks of computers as vast as we want them to be—in fact, it is reasonable to assume that virtually the entire fund of rational knowledge of the human species will be available to any of us well before the end of the 21st century. The limitation we face now is in designing access methods that will prevent us from being lost in the Palace—or worse, thinking we have found everything there is to know about some vital subject when we have only found what is nearby.

M. C. Escher: Mastering The Fluid Viewpoint

To aid us in our quest, we'll take a look at the work of the 20th century artist M. C. Escher. Escher was the master of integrating multiple viewpoints into a single work of art, enabling the user to become aware of the contextual nature of ordinary reality and break free of attachment to a single way of seeing.

Like Escher, we must empower our users with the means to shift viewpoint in the midst of navigation, skillfully and easily. We'll find a hint on how to do that by looking at the specific technique Escher used to provide his users with awareness of context.

Design Patterns For Information Architecture (An XML-Based Approach)

Building on these insights, we'll describe some design patterns we have developed for information architecture, and show how they are used to support modern hypermedia-based user interfaces such as a sophisticated pharmaceutical information Web site. We'll explore the challenge facing our team at Micromedex, a Thomson Healthcare company, as we work to bring its large, complex pharmaceutical knowledge bases into alignment with 21st Century business goals. We'll tell you what we've found out so far, and how we plan to use XML in our work.

Memory Palaces

History Of The Memory Palace

The concept of "memory palaces" recently re-emerged from medieval obscurity, surfacing in such diverse arenas as pulp fiction (Thomas Harris' *Hannibal*), conventional architecture (Lyndon & Moore's *Chambers For A Memory Palace*), and Internet architecture (Carl Malamud's article on Memory Palaces as millennial metaphor, found at <http://mappa.mundi.net/cartography/Palace/index.html>, which was a major inspiration for this paper).

The concept of Memory Palaces can be traced at least to ancient Greece and Rome. Aristotle, Cicero, and Tullius wrote extensive treatises on memory, both natural and artificial. The concept was practiced through the Middle Ages in the Italian scholastic monasteries. A tremendous revival occurred at the time of the Renaissance, ultimately derailed by the invention and dispersion of the printing press. A final resurgence of the concept occurred among traveling post-Reformation Roman Catholic missionaries. This history is much more fully elaborated in Frances Yates' *The Art Of Memory* and Jonathan Spence's *The Memory Palace Of Matteo Ricci*.

How Memory Palaces Work

A memory palace was a technique of memory enhancement in which a collection of linear knowledge spaces (speeches, epic poems, scriptures, etc.) was committed to memory by mapping the collection into an imagined physical space (either wholly imaginary or a remembered place in the real world). The practitioner associated portions of the work to be remembered with striking and unusual (therefore memorable) objects of various kinds placed in the imaginary space. Then by traversing the memory palace, viewing the objects in sequence and recalling the associated thoughts, the practitioner could reconstruct the work.

One of the most powerful aspects of the memory palace concept was that travel within such a palace was instantaneous, limited only by the speed of thought itself; hence related works could be accessed and interconnected in thought without limitation. One of the ancient treatises provided the following description of what sounds remarkably like a binary "chop-search" algorithm:

...the middle point also among all things is a good mnemonic starting-point from which to reach any of them. For if one does not recollect before, he will do so when he has come to this, or, if not, nothing can help him; as, e.g. if one were to have in mind the numerical series denoted by the symbols A, B, G, D, E, Z, I, H, O. For, if he does not remember what he wants at E, then at E he remembers O; because from E movement in either direction is possible, to D or to Z. But, if it is not for one of these that he is searching, he will remember (what he is searching for) when he has come to G if he is searching for H or I. But if (it

is) not (for H or I that he is searching, but for one of the terms that remain), he will remember by going to A, and so in all cases (in which one starts from a middle point).

- Aristotle, "On Memory And Reminiscence"

Benefits Of The Memory Palace Practice

As a practitioner of the art of memory, one became an invaluable asset to one's employer, an irreplaceable repository of knowledge that could otherwise only be stored and retrieved manually at great effort and expense. In a violent epoch, this asset value had tremendous survival value.

Furthermore, by interrelating to each other and to present experience the various works one had so carefully stored in one's memory palace, the practitioner could weave deeper and richer interpretations of current events than would otherwise be possible.

Modern Memory Palaces: The Age Of The Library

The single most significant invention of the past millennium was arguably the printing press, which effectively ended the need for prodigious feats of memory on the part of individuals. However, all new technologies have positive and negative aspects. The print medium imposed linearity on memory and recollection, reinforcing a structure that had first emerged with the invention of the written word at the dawn of civilization.

In spite of the expensive nature of its storage and retrieval methodologies, perpetuated by countless generations of scribes over the ensuing centuries, the written word successfully separated knowledge from the evanescent halls of memory palaces. A medieval library of fifteen or twenty thousand hand-transcribed, illuminated volumes arguably was as great a centralized aggregation of the fruit of human labor as could be found in the world, then or since.

The printing press revolutionized the reproduction of the written word, dramatically reducing the time and effort involved. It started an explosion of literacy that continued through the Reformation, the Enlightenment, the emergence and ascendance of industrial capitalism and democratic institutions, and up to the beginnings of the Information Age.

The print revolution created a new class of scholars who became, in effect, the inheritors of the Memory Palace tradition: the librarians. In the last half of the past millennium, library science evolved ever more elaborate schemes of classification for the concepts stored in their collections. Through diligent maintenance of their catalogs, thesauri, controlled vocabularies, taxonomic hierarchies, and cross-references, librarians created virtual pathways a seeker could traverse in the quest for understanding. The personal nature of the Memory Palace yielded to a community memory, embodied in its libraries.

Post-Modern Memory Palaces: The "Small e" Paradigm Shift

As often happens, only when a doorway appears on the horizon do we understand that we are imprisoned. With the emergence of electronic media in the latter half of the 20th

century, the limitation of the linearity of the printed word became painfully apparent. Marshall McLuhan's *The Medium Is The Message* marked the recognition point for the next revolution that would sweep the information scene. McLuhan believed that the linearity of the past was now over and that the print medium would soon die out. His predictions proved unwarranted in one sense—the print medium is at an all-time high in terms of sales—yet McLuhan clearly understood that non-linear media had tremendous potential and appeal that would transform the information scene.

At the same time as the linearity limitation became manifest, the means for its transcendence appeared, with the arrival of ubiquitous networking and the invention of hypermedia user interfaces. For the first time, institutions such as governments and businesses had the potential to place their entire organizational memory at the service of all parts of the organization: memory palaces were again a possibility, albeit at the corporate rather than the individual level.

Moreover, the possibilities do not stop at the organizational doorstep. As the Internet matures and the "small e" appears in almost every buzzword (e-content, e-commerce, e-publishing), the technical limits to the integration of disparate organizational information spaces are about to dissolve, not just for the few but for the many.

On The Threshold Of Unification

As a consequence, the world faces the prospect of the unification of virtually the entire fund of human knowledge. The Internet and the World-Wide Web represent this event in its infancy, and recent premonitions of the Semantic Web foreshadow its next evolutionary stage.

The World-Wide Web, first implemented in 1992, has strayed far from its initial scope and purpose. Begun as a way of connecting the ivory towers of academia scattered around the planet, it has instead become a richly textured marketplace of ideas and goods, coexisting in odd juxtaposition: lurid adult entertainment and hate literature; religious fanatics of every stripe, even atheist; commercial ventures ranging from multinational corporations down to remotely-located cottage industries; thousands of ordinary people suddenly offered the possibility of exposing their ideas to millions of others around the planet. And the millions will be billions soon enough.

A huge wave of interest and excitement surrounded the World-Wide Web for several years, a wave which lost some momentum recently as the NASDAQ stock exchange dipped sharply in early 2000. In spite of investor trepidations, however, it is clear that the Internet is here to stay as a communications medium. Whether the World-Wide Web *per se* continues to maintain its panache in the public eye, businesses and organizations have entered into a paradigm shift that will transition much of their communications infrastructure over to the Internet over the next several years.

Perhaps most importantly, large-scale information repositories are rapidly moving to expose their contents over the Internet. While the outward manifestation of this trend can be seen in consumer-oriented information provider sites such as <http://www.cnn.com> and <http://www.northernlight.com>, the real action is increasingly occurring at the business-to-business, business-to-professional, and organization-to-organization level. Synergies

between the many kinds of corporate and organizational knowledge bases offer possibilities that boggle the imagination, if only they can be successfully interlinked.

The Current Limits Are Technological And Political

So the Memory Palace concept is about to break through to a new level—species-wide coverage. We can see the evolution more clearly when laid out in tabular form:

| | Memory Palace Epochs | | |
|-------------------------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| | Classical | Modern | Post-Modern |
| <i>Scope</i> | Personal | Community | Species |
| <i>Manifestation</i> | Individual feats of memorization; iconographic art | Libraries | Digital libraries; specialized information repositories; the Internet |
| <i>Applications</i> | Rhetoric; passing along of oral traditions | Knowledge preservation; advancement of literacy | Knowledge preservation; automated knowledge creation |
| <i>Tools & Techniques</i> | Artificial and natural memory | Catalogs; controlled vocabularies; taxonomic hierarchies; thesauri; some automation in later stages | Ubiquitous, internetworked access to vast content bases stored in a variety of RDBMS/CMS systems |
| <i>Key Limitations</i> | "Threescore and ten years" | Degradation of physical media; cost of maintaining metadata manually; language barriers | Disparate formats of knowledge bases; proprietary ownership of information; governmental and corporate censorship |

As the scope of the Memory Palace moves from the person to the species, the limitations are changing as well. Immutable biological limits have been replaced by technical and organizational limitations, thorny problems to be sure but possible to overcome.

One key concept of the Memory Palace still rings true today: the separation of *knowledge* from *content*. In classical times, the structure of knowledge took the form of imaginary architecture and striking iconic imagery. In the realm of the modern library, knowledge manifests in the form of the catalog, with its indices by subject, title, and author, and with its cross-references in the form of "See" and "See Also" relationships. In the post-modern cyberspace Memory Palace, knowledge will take the a multiplicity of forms—full-text indices, topic maps, resource description catalog databases, and a variety of other structures we have only just begun to imagine.

Content continually grows and changes, but the structure of knowledge grows increasingly more stable as standardized vocabularies and domain-specific nomenclatures fall into place. There will always be a green edge where new insights are occurring, but the known universe of information will become increasingly civilized even as it expands exponentially.

M. C. Escher: Questioning The Assumptions Behind Perception

The Link Between Escher's Art And Knowledge Representation

The 20th Century artist M. C. Escher produced many drawings and engravings that challenge the assumptions behind the perception of ordinary reality. In many of his works, Escher challenged in particular the assumption that a single point-of-view is necessary for perceiving the world.

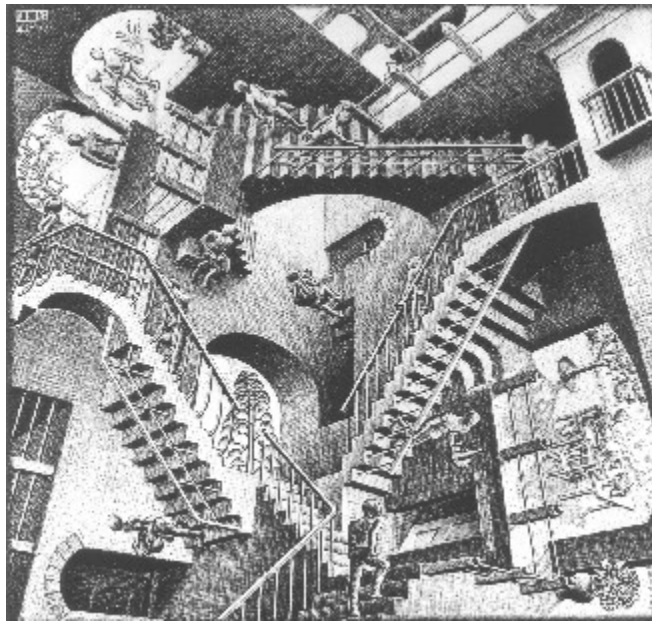


Figure 1. [Relativity, drawing by M.C. Escher, Copyright © by Cordon Art. All Rights Reserved.](#)

In works like *Relativity*, Escher demonstrated the apparently peaceful coexistence of several independent worlds operating on different, intersecting planes. A person walks within a strange building on staircases and down hallways, while another person coexists on a different plane only a short distance away, a plane in which the first person's walls are the second person's floor or ceiling. How many independent frames of reference can you count in this drawing?

Knowledge is exactly like this, and to see it in proper perspective it is very important that we shake up our conceptual viewpoint from time to time. Knowledge consists of the conceptual interconnections between the facts, definitions, analyses, opinions, and rebuttals that form the content; the more interconnections, the more knowledge. The more unexpected the connections we can create, the more creative and innovative the knowledge we can uncover.

So we must ask ourselves: what constitutes the knowledge equivalent of adopting Escher's multidimensional artistic approach? How can we best exploit the incredible diversity of the knowledge bases that are now or will soon be available on the Internet? As with Escher's drawings, the key is multiple points of view.

Choosing And Changing Points Of Reference

In drawing, a point of view is realized by choosing a "vanishing point," a point that forms the anchor for the frame of reference within which the objects are rendered. Objects in a drawing change size in direct proportion to their distance from the vanishing point: the smaller the distance, the smaller the object, until objects at the vanishing point simply *vanish*. The simplest example is a drawing of a road extending across a flat desert. The road at the bottom of the drawing appears wide and feels near; as it moves up the drawing the lines that form the edges of the road converge, meeting at the horizon, which constitutes the vanishing point in the drawing.

In his drawings, Escher often used multiple vanishing points to shake up the viewer's sense of direction. In several drawings, he employed the distorted reflection of convex curved surfaces, which effectively have an infinite number of vanishing points. He used every technique in his considerable repertoire forcing the user to question customarily unshakeable boundaries: figure and ground, foreground and background, artist and artwork, observer and observed.

Like Escher, we must empower the user of an arbitrarily large information space to shift smoothly and definitively from one point of view to another at will, allowing the user to see the same topic in the context of several different, even conflicting, frames of references. In the next few sections we'll look at how we can do this with some newly emerging technologies.

But the most important tool is not a technology: it's a paradigm shift that changes our view of the user from a passive spectator who merely accepts the information we serve to an active participant, a navigator in the information space.

Once again, the key insight we take from Escher's work is that the point of view—the frame of reference within which the objects of his art are located—is distinct from and exists independently of the objects themselves. This separation of framework and content, when applied to an information repository, has all the same advantages. We can easily empower the user to shift point of view smoothly in the context of the navigation of the information space—as long as our information architecture supports it!

Design Patterns For Information Architecture

Providing Support For Information Space Navigation

In our work, we find ourselves using certain kinds of tricks over and over again. They manifest outwardly in GUI features and inwardly in XML document types and their corresponding database schema structures, but they actually exist independently, sandwiched between these two layers, as heuristics for information architecture that repeat themselves over and over again.

Consistent with the rest of the software engineering world, we are working to expose and document these design patterns, and we have done so in considerably more detail at <http://www.swdi.com/design4ia/>, on the South Wind Web site.

The following table clarifies the separation of layers we are trying to describe:

| <i>Layer</i> | <i>Design Pattern Purpose</i> | <i>Manifestations</i> | <i>Standards</i> |
|---------------------------------|-----------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Application/presentation | User and programmatic navigation and control; knowledge visualization | Web Information Systems | HTML; PDF; RDF; SGML/DSSSL; XML/XSL; ArborText FOSIs for SGML or XML |
| Information architecture | Structuring the relationship between knowledge and content | Topic Maps; RDF Databases | ISO 13250; XTM (draft); RDF |
| Document/database design | Efficient storage and retrieval of information | RDBMS (e.g., Oracle, DB2, SQL Server) OODBMS (e.g., Documentum, Tamino, XPP, eXcelon) | XML; SGML; SQL |

The central theme of our work is providing support for **information space navigation**, and consequently we are intensely interested in the features of information structures that are required to support and optimize information space navigation. We work to define the mental models used as the core structures of all aspects of the information space, and

represent these models in the form of metadata that *represent* the mental models and *connect* them with the content in the information space.

Separating Knowledge And Content

Taken together, the mental models and the content connections form what we think of as the "knowledge layer." The mental models represent the abstract knowledge structures as they exist independent of any particular content base, and the connections to occurrences in content provide a mapping of this abstract knowledge to the current state of the fund of knowledge available to the user. We think this separation of knowledge from content is as important a step in the advancement of software engineering as the separation of data structures and algorithms that formed the foundation of the object-oriented (OO) movement.

We expect that regardless of where you find a window into a large information space—a well-designed news site on the Web, a workstation-based database search application, a well-designed PBX menu for an airline, or the controlled vocabulary interfaces used to search very large document bases like Westlaw or the Library of Congress—you will see these same patterns at work.

In our work and our examples, we focus on XML as an enabling technology for implementing these design patterns, and consequently we find ourselves looking closely at XML-based standards, such as Resource Description Framework (RDF) or the emergent XML Topic Maps standard (XTM) based on the SGML standard ISO 13250. This by no means implies that these approaches to implementation are the only ones available, and in fact user interface and database designers were using variations on these patterns for decades prior to the emergence of XML. However, we expect that other practitioners, like ourselves, will find that XML-based standards and tools increasingly dominate the marketplace of ideas in which these design patterns are a ubiquitous feature.

Design Patterns For Information Architecture

Here are some of the design patterns we use in our information architecture activities. They are grouped by functional activity.

Searching within the information space for material related to a topic

| Pattern Name | Description |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fuzzy Search | Provide methods of "fuzzy search" that can pull up subsets of topics that match the specified fuzzy search criteria. |
| Map of the World | Present to the user a map of the overall context in which the current topic is situated. |
| Intelligent Result Set | When queried for a topic of a particular type, return results organized around relevant subtopics. The topic type determines the set of relevant subtopics. The relevant subtopics may be organized as a list or a tree, depending |

| Pattern Name | <i>Description</i> |
|---------------------|-----------------------------------------------|
| | on their predefined topic interrelationships. |

Intelligent Sequencing

| Pattern Name | <i>Description</i> |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rating System | Arrange a sequence of objects in ascending or descending order according to some characteristic, which can be normative or objective. |
| Chronology | Arrange a sequence of objects in chronological order. |
| Tabs | Within a sequence of objects, bring out specific attribute values of every n th object, or the start and end attribute values of each subgroup of n objects, and use the extracted characteristics to mark entry points into the sequence. |
| Arcade | Tag each of a sequence of objects with identifying characteristics (like storefront signs in an arcade). |

Sharpening, softening, or recasting the focus of attention

| Pattern Name | <i>Description</i> |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Zoom Lens | Allow the user to increase or decrease the level of detail with which the current topic is viewed. Increased level of detail can be used as "blueprint" of the internals of the topic. |
| Facets | Bring into focus a particular subset of a topic's attributes. |

Transitioning to a related topic

| Pattern Name | <i>Description</i> |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Gestalt Shift | Move the focus of attention to a different topic within the current field of attention, effectively exchanging figure and ground. |
| Escher Staircase | Without changing the current topic, change the frame of reference within which the current topic is situated. |

Orientation and navigation within the information space session

| Pattern Name | <i>Description</i> |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Trails of Crumbs | While the user is navigating within a hierarchy or network of topics, leave markers that allow the navigator to jump back or forward to any point in the topics already visited. |

Thomson Healthcare's Challenge

Background

Micromedex began as a microfiche publisher in the late 1970's, cross-referencing household products, drugs and biologicals to their appropriate toxicologic management. The primary users of this system were poison control centers and hospital emergency rooms. The next product was the Drugdex® system, which is comprised of very large, highly structured monographs, presented in outline form, covering the complete range of clinical information relating to generic pharmaceutical substances. To the credit of the original system designers, these monographs were marked-up for content (e.g. section headers and indexable terminology) using a proprietary system of "dot codes" modeled after the IBM "Script" publishing product. This highly structured system allowed Micromedex to create one of the earliest PC based information products, the Computerized Clinical Information System (CCIS), in 1983 utilizing digital laser disc technology. This product was migrated to CD-ROM in 1985, making it one of the earliest CD-ROM offerings, and has been in continuous quarterly publication since that time. In 1997, CCIS was moved to a Web browser-based product (and renamed Healthcare Series). Many navigational features were added, especially an integrated index of databases and expanded hypertext links between databases. However, several key features were missing from these "legacy systems" that are crucial for moving into 21st century.

The Problem

Because monographs are authored as large, monolithic documents, the repurposing and reuse of data is very difficult. Creation of product subsets, mixing and matching documents for custom product delivery requires difficult custom programming. The lack of reusability causes the product authors and editors to duplicate information, causing serious maintenance efforts to maintain consistency across products. Data creation is often limited to end-product context, giving a "vertical" focus on the editorial process, rather than a "horizontal" knowledge-based focus. Finally, there is only limited use of controlled vocabularies and hierarchical lexicons. Searching therefore can return fragmented results sets, which are not navigable through use of lexical taxonomies. Integrated search results must be based on terminology values only and may give only partial results and force the user to try different search strategies.

The Solution

Micromedex is converting its monolithic monographs to small knowledge objects called *authoring units* [AUs]. Authoring units are heavily typed objects containing information based on a particular subject; for example drug-drug interactions, clinical uses, dosing, adverse drug events, pharmacology. AUs are described in XML and contain XML components – standard ways of describing certain types of information. These

components all move through the same authoring and review process, even if they are normally not published together. For example, a dosing AU may contain components for the MD, Pharmacist or Nurse, worded differently or contain different levels of specificity, but be essentially the same in content. These components would be edited together for consistency, but would be published in different products.

AU's are classed together in the Repository Hierarchy in order to facilitate searching and object inheritance. For example, all types of drug interactions, e.g. drug-drug, drug-food, drug-ethanol, drug-tobacco, drug-laboratory modifications, etc. all share common properties and are classed into the "Interactions" class.

Finally, to allow deterministic product assembly and searching, AU's store critical self-describing information as pointers into the drug and clinical lexicons, rather than terminology. For example, a drug interaction regarding aspirin would be coded to the drug generic master code of 1, rather than being stored as "ASPIRIN". This allows all the properties of the drug lexicon to be inherited by the authoring unit. These elements are "exposed" in the Repository index by attribute type. This is the primary method of searching the Repository.

Some requirements of an knowledge object based system

The solution to moving from monolithic documents to hundreds of authoring unit types with millions of instances requires a complete conceptual knowledge framework for editors, product developers, internal and external software developers, and the people who make it all possible, the physicians, pharmacists, nurses and healthcare professionals who are the Micromedex product users.

1. Editors must understand how to create data elements – what data goes where.
2. Editors must be able to find data based on various criteria – drug and disease names, data types, work queues, document status.
3. Software developers must have a simple model for querying the repository. There must be a consistent method for transporting data between software modules.
4. Software developers must have access to the properties associated with data types – indexed elements, workflow models, display attributes, etc.. They must have straightforward access to metadata associated with documents, e.g. workflow status and versioning information.
5. Product developers must be able to write rules to extract data from the repository and use it to assemble end user products.
6. End users must be able to navigate the Micromedex products using vocabulary that is familiar to their professional specialties. They must be able to find drugs by generic, proprietary and local names. They must be able to find diseases by broader and narrower terminology and synonyms.
7. External developers must have simple, understandable knowledge models for third-party implementations.

Architectural Components of the Micromedex Repository

The Micromedex knowledge repository is architected upon the following components.

1. Authoring Units [AUs] – Data objects are organized in the publishing system as *authoring units*. AUs have associated properties such as software access methods, editing tools, MIME type, exposed data elements, workflow model and DTD's. These properties are accessible in the Repository Hierarchy document.
2. Repository Hierarchy - The Repository Hierarchy is an XML object that provides the taxonomic hierarchy (classification system) for the retrieval of data elements (authoring units) within the Repository. AUs may themselves be considered as components of the Repository object. The Hierarchy associates AUs with their metadata properties. The Hierarchy also provides classification for non-textual elements (graphics, multimedia), lexicons, and third-party "flat-file" databases.
3. Data Index and Document Exposure – Defining elements of Authoring Units. Normally references to drug and clinical lexicons are exposed through the repository search subsystem.
4. Lexicon – Navigable taxonomies of drug and clinical terminology. Authoring units store references to lexicon and code translation tables rather than the term itself.
5. Product assembly rules – Rule templates to create end products as well as assist in the editorial creation of new documents.
6. The physical repository – Provides data object and metadata storage, workflow management, versioning, and multi-user access control.
7. Document Object Model (DOM) - Used to express tree-like data structures that comprise XML documents and rows and columns in relational tables. Provides encapsulation and messaging of XML document data and metadata.

The Implementation

Far and away the most complex part of the design of the Thomson Healthcare clinical knowledge repository (affectionately referred to as the Repository Of Clinical Knowledge or ROCK) is the classification and modeling of authoring units. Because of the wide range of information there are many areas of overlap. Defining these areas and modeling to reduce redundancy is sometimes a daunting task.

For example, Micromedex displays drug dosing in nearly 40 different contexts. Analysis shows that it will take at least six, and possibly eight AUs to maintain current functionality. Creating eight drug dosing AUs leads to the problem of ambiguity when creating new data. This is where the classification scheme of the Repository Hierarchy becomes of critical importance. The ability to navigate seamlessly and fluidly through the various AU types allows the editor to maintain perspective.

Tagging data, which may have formally lived as untagged text, is an expensive task. Understanding the proper level of tagging granularity has a major impact on overall editorial investment in the migration of data. Migrating clinical indexing that currently

exists to a standard lexicon, such as SNOMED RT, requires a massive editorial reconciliation effort and entails multiple techniques—including taking long-term approaches.

The Thomson Healthcare clinical knowledge repository utilizes the Documentum content management system to provide many required services—workflow, versioning and system maintenance.

The Document Architecture

The Micromedex Document Architecture is an organizing scheme for creating an implementation hierarchy for authoring units. Authoring units are related clinical information grouped into logical units allowing editors to create and modify data. In addition, authoring units have the ability to store pointers to controlled information (lexical, relational) and have meta-data associated with them.

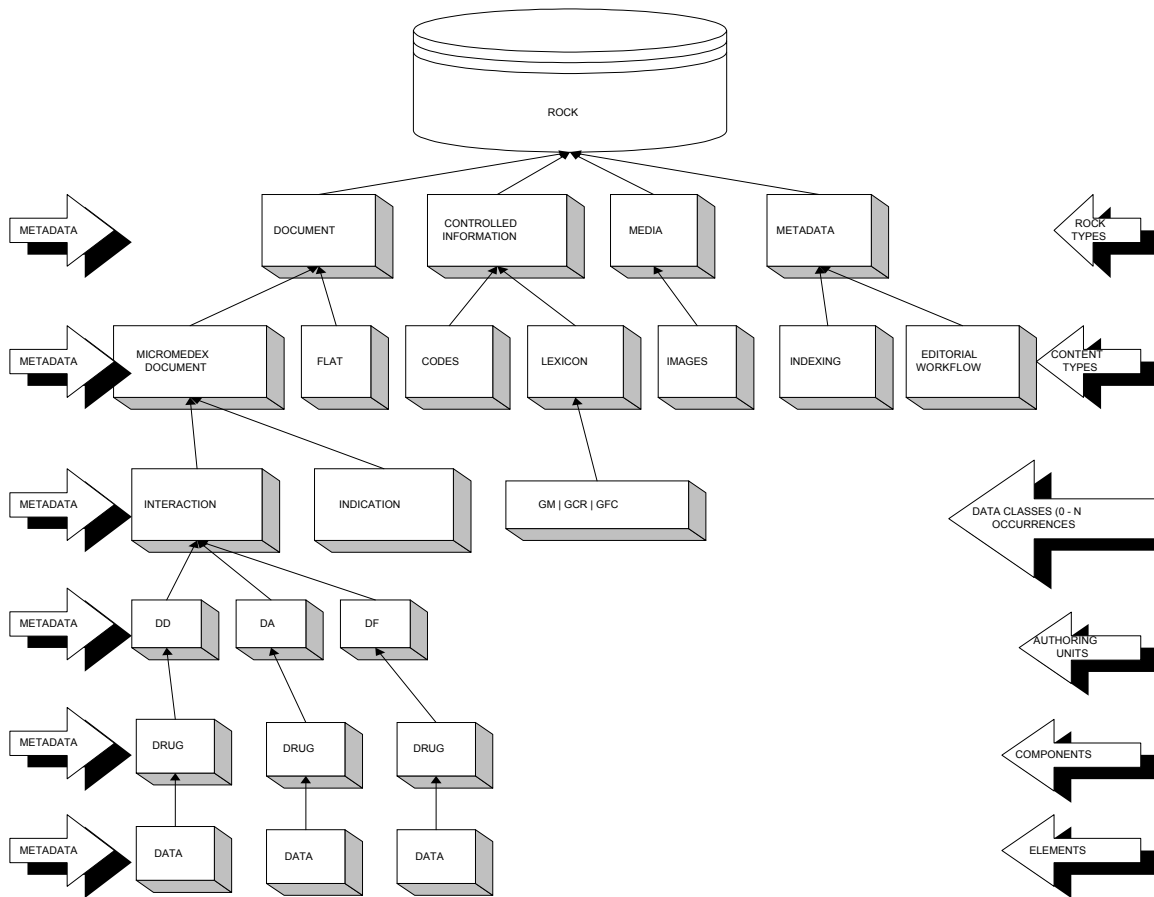
Two important concepts from the Document Architecture are the *Repository Hierarchy* and the *Hierarchy Access Notation*. These are briefly described in the following subsections.

The Repository Hierarchy

The Repository Hierarchy is a virtual XML object that provides the taxonomic hierarchy for the storage of data elements within the Repository. The ROCK is an instance of the Repository Hierarchy. Each and every accessible Authoring Unit, media type, third party document type and other data types are explicitly named in the Hierarchy. Data types and their properties, accessor methods, etc., will be accessible utilizing the Hierarchy Access Notation, briefly described below.

Because the Repository Hierarchy is an implementation hierarchy, Java class inheritance can be used to implement accessors. Using Java's Reflection capabilities, our accessors are server-based objects that leverage the enormous power of our back-end hardware. This will allow us to deliver functionality on virtually any client platform.

The accompanying drawing gives a very sketchy picture of the overall hierarchy, which is far too large for a single drawing to do it justice.



The Hierarchy Access Notation

While outside the scope of this discussion given the limited time available, it is worth mentioning that our implementation provides an access method for describing the position of and accessing the contents of every element in the Repository Hierarchy. We call this access method the Hierarchy Access Notation, or HAN.

Every data type that is stored in the Repository is expressed in terms of its hierarchical position using “dot” notation. Hierarchy Access Notation is used to specify a particular named node in the repository. Each node, from highest level to lowest level is specified in left to right order, separated by a period (“.”). Because data type names are unique, the notation can be *minimized*, allowing only data type (e.g. the leaf node such as an Authoring Unit) and components to be explicitly stated. HAN describes a node in the document hierarchy; a node may consist of a document (authoring unit) or a document component (an addressable element within an authoring unit, e.g., a subtree within an AU's XML structure).

Document Instance Notation is defined to allow the specification of a single instance or set of document instances. HAN only describes a document instance (or instances, since it can describe a set) when combined with a DIN component. As it happens, the HAN notation is structurally similar to XPath notation, but the HAN notation was developed independently to serve Micromedex-specific purposes within the ROCK.

Topics And Occurrence Links

The Document Architecture's lexicon subtree neatly encapsulates the rough equivalent of the Topic Map concept of "topic." The Document Architecture's default hierarchy provides a means for doing the Topic Map equivalent of associations, and additional associations can be created de facto using occurrence links within content documents. We have learned that some types of content documents (for example drug formulations) are almost entirely defined as a nexus of links to other topics (in the case of drug formulations, links to the component drug compounds, indications, side effects, interactions in which the formulation participates, etc.). Hence, these kinds of content documents represent associations as well.

The question of occurrence links is more difficult, since the traversal of such links constitutes the main potential performance bottleneck. We have adopted an approach based on the core technologies of the repository: Documentum, the Verity full-text search engine, and Oracle. Together these technologies represent a state-of-the-art implementation of the Fuzzy Search algorithm, so one base is already covered.

Now we face a key architectural question: can we implement all the other information architecture design patterns for our publishing system using variations of the fuzzy search design pattern?

Can "Fuzzy Search" Do It All?

Our occurrence links will rely on embedded structured tags in the content, tags that might look like the following:

```
<DRUGCODE SRC='LEX.DRUG' TYPE='GM' CODE='1'>ASPIRIN</DRUGCODE>
```

In this case, the SRC='LEX.DRUG' indicates the use of our drug lexicon, and TYPE='GM' indicates that we are referring to the "generic master" subtree, a list of individual drug compounds. The code attribute indicates which entry in the table is being referenced. The most common display name for the generic is the content of the tag; alternative names such as "acetylsalicylic acid" can be retrieved via the lexicon, and if embedded in a content document would have the same tag wrapper.

The start tag of the DRUGCODE element forms a deterministically dependable lookup key in full-text search. These tags are added by our editing software, never edited by hand, so the content of such a tag is always character-for-character identical for tag pairs with identical meaning.

Why is full-text search for structured keywords a better approach for us than maintenance of a separate index of topic occurrence links? First, we believe that one tool is generally better than two or more, and we already have a full-text search mechanism that kicks serious booty. Second, this will enable the editors to maintain links in context rather than having to use a separate linking tool. This makes it more likely that indexing will get done correctly. Third, though performance might be noticeably better with a separate index, we expect this method to perform acceptably for the editorial environment. Our products use a separate set of proprietary indexing

mechanisms generated from the actual content of any given product, and we do not plan on changing that for the foreseeable future.

To complete the picture, our Fuzzy Search entry points must provide a seamless thesaural lookup from lexicon terms to structured tags. The following sections describe at a very high level the algorithms we can follow to implement each of the design patterns using full-text search on embedded keyword structures.

Searching within the information space for material related to a topic

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fuzzy Search | <p>This is easily implemented using full-text search, and is nicely augmented by a thesaural search capability based on the embedded structured keyword approach.</p> <ol style="list-style-type: none"> 1. Search is performed against the full-text index. Thesaural translation from natural-language terms to structured embedded keywords serves as a way of ensuring that relevant topics are located. Returned results include document identifier and descriptive text. 2. The system prioritizes results by ensuring that keyword-tagged documents are placed ahead of simple occurrences of the natural-language term(s). 3. Result sets are returned to the user as a VDOM¹. |
| Map of the World | <p>There are a lot of different possible "World Maps" for any complex knowledge base. For our repository, we offer a default map—our Document Architecture—which provides direct hierarchical access, implemented in the repository storage schema, to the most commonly requested view of the authoring units, a functional viewpoint.</p> <p>The lexicons provide alternative "World Maps," and it is possible (albeit very likely less efficient) to use structured full-text search to traverse lexicon-based hierarchies. Authoring units reside in one subtree of the Document Architecture, lexicons as branches in another subtree. In either case (Document Architecture or lexicon hierarchy), the approach is the same.</p> <ol style="list-style-type: none"> 1. A VDOM implementation is used to step through the World Map. The VDOM wraps whatever hierarchy the |

¹ VDOM = Virtualized Document Object Model (VDOM). A VDOM looks to an application like any other DOM interface set. An application accesses the VDOM as if it were an in-memory object. When attempting to access a NodeList that is not yet in memory, access is blocked until the NodeList can be populated. Memory is managed using a standard garbage collection algorithm. The VDOM is a Micromedex extension to the Documentum access methods.

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>application desires, whether the default hierarchy or an alternative lexicon-based hierarchy.</p> <ol style="list-style-type: none"> <li data-bbox="573 380 1365 449">2. The application chooses how much or how little to expose of the World Map encapsulated in the VDOM. |
| Intelligent Result Set | <ol style="list-style-type: none"> <li data-bbox="573 470 1321 539">1. Perform a full-text search against the repository for the structured equivalent of the topic keyword. <li data-bbox="573 558 1377 695">2. Organize the topic results by data type (as the term is used within the Repository Hierarchy). Present the results to the user as a VDOM. This gives a context to the keyword occurrences. <p data-bbox="524 720 1360 863">Example: a search on Aspirin will return results for 'comparative efficacy, drug-drug interaction', 'adverse effects', etc. The results are organized in the same way as the encompassing subtree of the Repository Hierarchy.</p> <p data-bbox="524 884 1365 1056">For the purposes of our Publishing System effort, this design pattern is expected to be the most commonly used and versatile of the ones presented here. Many of the patterns presented here are actually more appropriate for inclusion in data sets extracted from the repository for the purpose of building products.</p> |

Intelligent Sequencing

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rating System | <ol style="list-style-type: none"> <li data-bbox="573 1241 1365 1377">1. Obtain document fragments using full-text search on structured keyword patterns. Fragments contain lookup key for entire document, descriptive text element, and element containing rating basis data. <li data-bbox="573 1398 1383 1472">2. Place document fragments into stream ordered by rating basis data. Return to application. <li data-bbox="573 1493 1365 1562">3. After application selects topic, create VDOM wrapper based on default hierarchy (the Document Architecture). |
| Chronology | <ol style="list-style-type: none"> <li data-bbox="573 1587 1365 1724">1. Obtain document fragments using full-text search on structured keyword patterns. Fragments contain lookup key for entire document, descriptive text element, and element containing chronological data. <li data-bbox="573 1745 1227 1818">2. Place document fragments into stream ordered by chronological data. Return to application. <li data-bbox="573 1839 1365 1896">3. After application selects topic, create VDOM wrapper based on default hierarchy (the Document Architecture). |

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tabs | <ol style="list-style-type: none"> 1. Obtain document fragments using full-text search on structured keyword patterns. Fragments contain lookup key for entire document, descriptive text element, and element containing tabbing basis data. 2. Place document fragments into stream ordered by tabbing basis data. Return to application. 3. After application selects topic, create VDOM wrapper based on default hierarchy (the Document Architecture). |
| Arcade | <ol style="list-style-type: none"> 1. Obtain document fragments using full-text search on structured keyword patterns. Fragments contain lookup key for entire document, descriptive text element, and element containing collation key for the document. 2. Place document fragments into stream ordered by collation key. Return to application. 3. After application selects topic, create VDOM wrapper based on default hierarchy (the Document Architecture). |

Sharpening, softening, or recasting the focus of attention

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Zoom Lens | <ol style="list-style-type: none"> 1. A VDOM wraps the hierarchy that represents the application's current frame of reference, whether the default hierarchy or an alternative lexicon-based hierarchy. 2. The application uses the VDOM to increase or decrease the level of detail within the current topic by exposing a greater or lesser number of DOM layers. Application can use greater layer exposure as "blueprints" of the topic's internals. |
| Facets | <ol style="list-style-type: none"> 1. A VDOM wraps the hierarchy that represents the application's current frame of reference, whether the default hierarchy or an alternative lexicon-based hierarchy. 2. The application uses the VDOM to choose among predefined subsets of elements within the current topic, subsets that comprise the different facets of the topic. <p>Facets are closely related to the Escher Staircase pattern; in fact, the Escher Staircase pattern can be implemented as a combination of Facets and Gestalt Shift.</p> |

Transitioning to a related topic

| Pattern Name | Implementation With Full-Text Search On Embedded Keyword Structures |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gestalt Shift | <ol style="list-style-type: none"> 1. A VDOM wraps the hierarchy that represents the application's current frame of reference, whether the default hierarchy or an alternative lexicon-based hierarchy. 2. From the document content as expressed in the VDOM, the application chooses an alternate topic and shifts to that topic, keeping the current frame of reference. <p>For example, from a drug-drug interaction document, choose one of the drugs and make that the current topic within the Document Architecture hierarchy.</p> <p>Gestalt Shift is closely related to the Escher Staircase pattern; in fact, the Escher Staircase pattern can be implemented as a combination of Facets and Gestalt Shift.</p> |
| Escher Staircase | <ol style="list-style-type: none"> 1. A VDOM wraps the hierarchy that represents the application's current frame of reference, whether the default hierarchy or an alternative lexicon-based hierarchy. 2. Using the identifier of the current topic, find other lexical hierarchies in which this topic is also situated, and present them to the application. 3. Allow the application to choose a different frame of reference from this set. 4. The selected hierarchy becomes the new content of the VDOM, with the current topic (unchanged) determining the current positioning in the VDOM. <p>The Escher Staircase pattern can be implemented as a combination of Facets and Gestalt Shift.</p> |

Orientation and navigation within the information space

| Pattern Name | Description |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Trails of Crumbs | <p>This is tricky to implement, but possible.</p> <ol style="list-style-type: none"> 1. A record must be kept of the current frame of reference and current topic for each of the past positions. The network of relationships traversed to reach each point must also be recorded, since there may be multiple ways to move from point to point. The network is made available to the application as an XML document or VDOM. 2. The application then chooses any point in the trail and makes that point the current position again. Moving in this |

| Pattern Name | Description |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>manner should not alter the trail's overall structure, since it is a conscious move to a prior position.</p> <p>Because it is expensive to implement, this pattern is not often found, but it has great utility from a user's perspective.</p> |

XML Approaches To Memory Palace Construction

Possible Approaches

There are several possibilities for the sort of complex metadata descriptions we need to create, though for the near term we will rely on our proprietary implementation until the technology dust settles. The following are brief descriptions of our leading long-term candidates. We elected not to expand further on these topics because they will be well covered in other talks in this symposium.

- XML Topic Maps (XTM)
- Resource Description Framework (RDF)

XTM

XML Topic Maps are an outgrowth of the SGML-based ISO 13250 Topic Maps standard. Where ISO 13250 relies on SGML for its syntax and HyTime for its document linking standard, XTM will use XML for syntax and XLink for its linking mechanism. ISO 13250 uses the HyTime concept of the Bounded Object Set (BOS) to interweave multiple SGML documents into a topic map, and XML does not yet have a fully formed equivalent to the BOS. The standard is still very much in flux, but it shows great potential.

The basic tenet of the Topic Map approach is very much in keeping with our work: the independent value of the knowledge layer (topics and topic associations) distinct from and in addition to the value of the content, and the consequent separation of knowledge and content in the implementation. There are some ISO 13250 features, like Facets, that do not seem well thought through, and we are hoping the XTM effort will find a more flexible and extensible approach to attaching metadata that is specific to individual content resources.

RDF

RDF's approach assumes an existing content base, and creates an external mapping of property names and values to resources. This approach works well for our situation. While RDF is conceptually well suited, our impression of the initial specifications is that they were made needlessly complex; if the complexity suits some need it is not apparent to us, and we hope a "Simple RDF" movement will arise akin to the "SML" and "Common XML" efforts around the XML specification itself.

Conclusions

We are at an exciting juncture, both in the industry at large and in our work. We see ourselves on the front lines of a revolution that is taking place across the entire world of information providers. Much remains to be done to unify our repositories and bring them into the 21st century. We look forward to the challenges we face.

More challenging is the effort the industry at large faces, unifying a multitude of disparate repositories, many of which are not yet even available in electronic form. As more and more original source materials from the world's libraries are digitized and brought online, the option of doing primary resource will no longer be an academic privilege. As exciting as the World-Wide Web is in its current form, it must be recognized as a very limited and biased slice of the human fund of knowledge.

Synergistic interconnections between the myriad repositories will only be possible as standards emerge and are accepted, and moreover when adherence to them becomes as simple as using a word processor is today. (We do not say "as simple as using an HTML editor" because the available HTML editors are not even close to the current generation of word processors in power and simplicity.)

In our work we are free to choose either to follow or ignore the nascent XML-based information architecture standards, since we are building purely internal systems, and our current decisions are based primarily on how the standards will affect our short-term productivity. This freedom is important in maintaining our competitive position in a very demanding marketplace. It is also important because the structures we build must be stable, and the standards (as this paper is written) do not provide an adequate level of stability and power to meet our needs.

For the present, we strive to ensure that we can generate the structures the new standards dictate, once they are fully solidified and support for them becomes imperative for interoperability with our vendors and customers. In the long run, however, we will build even our internal systems from the ground up for interoperability. The benefits are undeniable. Just as we now use TCP/IP and HTTP without question in both our internal and external systems, we expect that some day soon we will be using Topic Maps, RDF, or some similar standard for our architectural design, simply because it is the best possible all-around choice. As the information we provide is integrated into ever growing, multi-vendor distributed systems, such standards will ease the pain of everyone involved in development and deployment of the systems.

About The Authors

Leonard Rann has worked in the field of health information systems for the past twenty-three years, most of the time at Micromedex. In his role as Principal Systems Architect, he has kept Micromedex at the cutting edge of electronic publishing technology since the term was first invented.

Dale Hunscher has worked in the IT industry for 17 years, developing widely-used commercial products in the database connectivity arena for more than eight years as CEO of South Wind. For the last two years, in partnership with Connectex, LLC, he has consulted on the architecture of large scale XML-RDBMS and XML-OODBMS document management systems. South Wind's clients have included Pervasive Software Inc., Bell & Howell Information & Learning, and the Thomson Corporation.

References And Recommended Readings

Memory Palaces

Harris, Thomas, *Hannibal*, Dell Publishing, a division of Random House, New York, NY, 1999.

Lyndon, Donlyn, and Moore, Charles W., *Chambers For A Memory Palace*, The MIT Press, Cambridge, MA, Second Printing 1997.

Malamud, Carl, *Memory Palaces: A Millennial Metaphor?*,
<http://mapa.mundi.net/cartography/Palace/index.html>, 1 January, 2000.

Spence, Jonathan D., *The Memory Palace Of Matteo Ricci*, Penguin Books USA, New York, NY, 1984.

Yates, Frances A., *The Art Of Memory*, the University of Chicago Press, Chicago, IL, 1966.

Knowledge Representation And Information Architecture

Kulikauskas, A., of Minciu Sodas Laboratory, and Maskeliunas, S. of the Institute of Mathematics and Informatics, both of Vilnius, Lithuania: *Organizing Thoughts Into Sequences, Hierarchies, and Networks*.

M. C. Escher

The official M. C. Escher Web site is at <http://www.mcescher.com>. Reproduction of *Relativity*, a drawing by M. C. Escher, in this paper was by permission of Cordon Art, B. V., Netherlands. All rights reserved.

XML-Related Design Patterns

We highly recommend Toino Lainevool's excellent XML patterns site:
<http://www.xmlpatterns.com>

Information Architecture

Unfortunately, most resources on information architecture focus on the visible (and hence most popular) aspects of the work, namely the structure of Web sites. Information architecture as a discipline precedes and far exceeds the scope of the World-Wide Web. Nonetheless, valuable general knowledge can be derived even from the available resources focused on Web-related issues. The following books and sites are some of our favorites.

Morville, Peter, and Rosenfeld, Louis, *Information Architecture For The World-Wide Web*, O'Reilly & Associates, Cambridge, Massachusetts, USA, 1998.

Louis and Peter's company, **Argus Associates**, has a Web site with a valuable guide to information architecture resources: <http://argus-inc.com/iaguide/index.shtml>.

The **D-Lib Forum** (D-Lib stands for digital libraries) has a lot of valuable information, both academically and practically oriented: <http://www.dlib.org/>

Wired Magazine's Web site has an area called **WebMonkey** that is focused on tools and techniques for Web development. They have an information architecture tutorial (utterly Web-centric but still entertaining and useful) at http://hotwired.lycos.com/webmonkey/design/site_building/tutorials/tutorial1.html.

The **Usable Web** site's Topics page (<http://usableweb.com/topics/>) has a link that performs a search on information architecture material available through their site.