Managing Legal Document Collections:
The Use of Information Modeling
to Understand Intellectual Contents*

HAIM KILOV

Understanding and modeling document collections was successful long ago, before any computer systems became available: good reference librarians modeled collections of traditional documents in their heads (and notebooks!) and successfully used these models. Their approach is based on abstraction and precision – the currently recognized foundation of object information management [Kilov, Ross 94; ODP 2]. Indeed, a reference librarian – a taxonomist – abstracts away irrelevant details, such as layout and presentation, when building and using a model of documents in the library. Different contents details may also be considered irrelevant, depending upon the modeled viewpoint. Furthermore, the reference librarian’s model is precise, so that it is possible to answer questions like “do you have information about XYZ and if yes, where can I get it?”. In other words, a model of this kind is essential to help the user who is not interested in a fancy document, but rather is interested in solving a business problem.

Unfortunately, all too often such models are not presented explicitly. With a drastic increase in the amount of available information, the absence of explicit models becomes a serious problem: a reference librarian is not

* Copyright © 1994, Bell Communications Research, Inc. (Bellcore). Permission to use, copy, modify and distribute this material for any lawful purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies, and that the name of Bellcore not be used in advertising or publicity pertaining to this material without the specific, prior written permission of an authorized representative of Bellcore. BELLCORE MAKES NO REPRESENTATIONS OR WARRANTY, EXPRESS OR IMPLIED, ABOUT THE ACCURACY, SUFFICIENCY, OR SUITABILITY OF THIS MATERIAL FOR ANY PURPOSE. IT IS PROVIDED “AS IS”, WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. Bellcore expressly disclaims any liability for any damage or injury incurred by any person arising out of the sufficiency, accuracy, or utility of any information contained herein. Any use of this material is at the sole risk of the user.

Stanislaw Ulam describes abstraction – essential for a theoretical physicist – in his Preface to George Gamow’s autobiography [Gamow 70] as “gift for succinct formulation and ... ability to observe the essence of a matter through a mass of details”.

always available, and information management systems are useful only to
the extent they help the user understand and retrieve the information he
needs to solve a business problem.

Different users may have different viewpoints on a document collection:
they are interested in different collections of information fragments. In
other words, the contexts within which the users solve their problems are
quite different and do not always correspond to the contexts within which
the document authors have created their documents. Thus, the document
models created – implicitly or explicitly – by document authors may be
quite different from document models needed by document users. This
phenomenon is well-known to, e.g., students who highlight fragments of
textbooks needed to pass a test; or to attorneys who highlight those existing
document fragments that they need to support their arguments.

Obviously, understanding a document collection should not depend on
its presentation (paper or electronic). Electronic document management
may provide additional functionality (such as keyword search) to document
users, but should be able to provide at least the same functionality as
existing traditional document management. This has not always been the
case: existing approaches to understanding electronic documents have too
often – explicitly or implicitly – been based on existing tools. These tools
usually underemphasized document contents and overemphasized layout
and representation. As an example, consider highlighting by paper document
users: how many electronic document management systems support user-
created document models?

Keyword search looks like a useful content-related electronic document
management facility. However, its suitability for contents specification is
rather limited. Consider, for example, the same document content element
represented differently in different natural languages\(^2\); or for different
audiences (legal specialists or laymen)\(^3\). In all these cases, the same concept
is represented by means of very different keywords, and there is no
guarantee that the document author’s keyword will be equal to the
document user’s one. Other approaches, such as using SGML, succeed in
separating document’s physical presentation from logical layout and
contents, but do not clearly separate contents from layout. In addition,

\(^2\) George Gamow [Gamow 70] refers to the French and German editions of the Bible
which look entirely different in print, but are identical in content.

\(^3\) A name, including a keyword, is usable for denoting entities only within a particular,
explicitly specified, context [ODP 2]. The same entity may therefore have different names in
different contexts.
they are applicable to restricted classes of documents, mostly hierarchical. Moreover, many database management systems treat documents just as “unstructured data”. Document users were less than happy with these approaches and requested a better framework to deal with documents.

Highly qualified document users (attorneys, reference librarians, tax accountants, and so on) successfully use document contents to look for document fragments they need. They use document contents specifications implicitly: in human-to-human communication you can always ask additional questions and you can rely on a possibly implicit context. In computer-based information systems you cannot do that: a system has to be instructed precisely and explicitly. Therefore explicit document contents management is essential for solving electronic document management problems. Fortunately, there is no need to reinvent document management concepts from scratch: document management is no different from managing any large and complex application. Therefore the same information modeling concepts that have been successfully used in specifying other businesses, such as telecommunications, can be successfully used in specifying the business of document management. This paper will provide a short overview of information modeling in document management, and will show an example—a collection of legal documents (disclaimers) that has been successfully understood, modeled, made mutually consistent, and used.

Let Us Use Information Modeling

Understanding the semantics of a document collection can be accomplished using concepts and constructs needed to understand the semantics of any other enterprise [Kilov, Ross 94; ODP 2]. These concepts and constructs described in information modeling are independent of any particular methodology or tool. They may be used to create an abstract and precise model—the specification—both of an application and of documents that describe the application. Such a specification will explicitly describe the semantics of documents (concept maps) and assure interoperability between documents belonging to different collections.

The concepts and approach used in document management are applicable to any collection of documents: be they paper or electronic; “linear” or hypertext ones. The intellectual contents of a collection of documents does not depend upon its layout or the presence of hypertext tools. Therefore we need to clearly separate document management concepts from implementation technology. This will permit us to concentrate on the intellectual
contents of documents independent of how they are logically or physically represented.

A typical document is an ordered Composition of non-decomposable "elementary units" of information. Composition has to be described precisely. An elementary unit is exemplified, depending upon the viewpoint, by a character, a graphic unit like a line or circle fragment, or a sound unit; by a paragraph or a footnote; by a simple concept. But we also need to know what a Composition is, and a warm and fuzzy feeling is obviously not a definition. "Composition is just a whole-part relationship" is an example of such a warm and fuzzy feeling. Fortunately, a precise definition of a generic Composition exists and can be reused. It is based on those properties of a Composition that do not change (remain invariant) [Kilov, Ross 94; GRM 94; US Position 94], no matter what operations are applied to the Composition and its elements: A composite type corresponds to one or more component types, and a composite instance corresponds to zero or more instances of each component type. There exists at least one resultant property of a composite instance (dependent upon the properties of its component instances). There exists also at least one emergent property of a composite instance (independent of the properties of its component instances). The sets of application-specific types for the composite and its components should not be equal.

As an example, consider a submitted paper (a composite type) as a composition of an abstract, several sections, and a reference list (component types). We can imagine a submitted paper without an abstract or without a reference list, so that for these component types the number of instances may well be zero. Furthermore, there exist resultant properties of a submitted paper, such as the number of pages, or, to be less trivial, acceptability of the submitted paper to the journal. There also exist emergent properties of a submitted paper, such as the author, the title, and so on. Finally, the type "submitted paper" differs from the types "abstract", "section", or "reference list".

A Composition can be subtyped using several mutually orthogonal criteria:

- serializability – whether the components are ordered or non-ordered;
- changeability – whether the dynamic entry/departure of components is possible after the Composition has been established;
- hierarchy – whether a component can be associated with more than one composite;

4 On a lower abstraction level, we may note "uniformity of representation... as a bit stream" [Jain 94].
linkage – whether the component or composite can exist independently of the existence of the Composition association.

In a submitted paper, the components are ordered, changeable and non-hierarchical (the same picture, for example, may be reused in several papers). Both the paper and its components may exist independently of the existence of the Composition association.

The Composition concept may be reused in different document management viewpoints. Indeed, a document element may be viewed from three mutually orthogonal viewpoints: the physical presentation viewpoint, the logical layout viewpoint, and the contents viewpoint. (We have seen that a document element may be exemplified as a character, or a footnote, or an idea.) A document user creates composites out of these elementary document elements and works with (in the simplest case, reads) these composites. Each elementary unit and therefore each composite of elementary units may have several different physical representations. Different document users are interested in different aspects of a document’s contents and therefore may, and usually do, combine document components differently. A document – or a document collection – is not only a hierarchy: different users use different components of a document collection, both those outlined by the author, and those discovered by the user. When an attorney researches legal documents to support his viewpoint, he usually creates new documents out of various components of existing ones. When a university professor prepares course materials for students, he often reuses various fragments of existing textbooks, of journal papers, of conference proceedings, of viewgraphs, and so on.

The Big Three Viewpoints

Let us describe the document management viewpoints in more detail. We will clearly distinguish between the following big three viewpoints:

- physical presentation
- logical layout
- intellectual contents.

From the physical presentation viewpoint, a document may be considered as an ordered Composition of pages (or page images on a screen), and

5 These viewpoints may be mapped onto the Open Distributed Processing viewpoints [ODP 2].
a page as an ordered Composition of lines. From the logical layout viewpoint, a document is an ordered Composition of chapters, and a chapter is an ordered Composition of sections, whereas a section is an ordered Composition of paragraphs (obviously, pages and lines are not dealt with here). A more structured document, such as a book, may be considered as an ordered Composition of a front matter, followed by a table of contents, followed by a sequence of chapters, followed by a reference list and an index. A document with a given logical layout will have many possible physical presentations, depending upon the font, the font size, the page or window size, the distance between paragraphs, and so on – and we have said nothing about non-textual document fragments!

Finally, and most importantly, a document (collection) is used to convey contents. In simple cases, document authors present their document contents models in a clearly visible way, using logical (and possibly physical) layout. Well-known examples include annotations (presented on paper as boxes, footnotes, highlighting, and so on; presented electronically as buttons, separate windows, and so on), warnings, etc. However, document contents elements need not coincide with existing document logical layout elements (and obviously, with document physical layout elements). A document contents element may correspond, e.g., to several words in a paragraph, or a collection of paragraphs, or a paragraph and a half, or a section, or a fragment of a picture, and so on (consider document contents elements highlighted by students in their textbooks!). The logical (or physical) layout of a document may be changed (improved), without changing its intellectual contents.

Let’s try to formulate a specification of the most important aspects of a collection of documents. Document contents elements should be “visible”. This is necessary in order to “point to” (refer), reason about, and exchange these elements: there is no need to deliver a document to a user who is interested only in a small part of the document! The logical layout of a document is used to make document elements visible, and so there should be a logical layout element (e.g., a box, a picture, a footnote, a chapter, a highlighting, an index entry, and so on) for each contents element. Some of these elements – like highlighting or perhaps index entries – are created by a document user, especially if the user’s intellectual contents viewpoint differs from the document author’s one. Obviously, there is no need to have a contents element for each and every logical layout element. Therefore

---

6 Compare with anchors in Dexter and link markers as “permanent ties” in its DHM extension [Grønbaek 93]: link markers can track editing changes to the component’s contents.
only some logical layout elements correspond to contents elements, and there is a need to appropriately subtype logical layout elements. We obtain the following (precise and abstract) specification:

A visualization logical layout element is used for visualizing a document contents element (to be explicitly shown in the document itself), so that there exists a Reference association between the two. It means that some properties of the visualization element (the maintained entity) correspond to the properties of the documents contents element – its reference entity. The Exhaustive and Overlapping Subtyping ($SE+$) and Reference associations are rigorously defined in [Kilov, Ross 94] in the same manner as the Composition association above. An instance of a document logical layout element will belong, in the contents-oriented exhaustive and overlapping subtyping hierarchy shown above, to at least one of its subtypes (visualization or traditional); however, other subtyping hierarchies for a document logical layout element are also possible. They are not shown here.

The graphical representations of relationships shown above are by no means the only ones possible. For those who do not like cartoons, an equivalent linear representation may be preferable [Kilov 94]:

Contents-related: Exhaustive and Overlapping Subtyping (Document logical layout element, [Traditional, Visualization])

Contents Visibility: Ordinary Reference (Document contents element, Document logical layout element)

A document (collection) is used to convey contents. We can use the same concepts in modeling a collection of books and a collection of electronic documents. However, there exists a difference between books as (say) collector's items (first editions, art books and so on come to mind...) and books [and journals] as collections of fragments of interest to the reader.
When we reuse the intellectual contents of books (e.g., create our own collection of documents for this purpose) we obviously consider books in the latter sense. Therefore we often do not include in our collection a complete document, but include only those fragments that we need. We as document users wish to define the granularity level ourselves, rather than deal with complete documents or always rely on document authors for defining the granularity level for us.

A document user may browse and understand the document using its implicit or explicit model provided by the document author. Ideally, a document model (concept map or several concept maps) should be shown explicitly, by presenting appropriate information specifications to the users. The authors of a document may present their viewpoint of this concept map in a clearly visible way, using the document’s logical (and possibly physical) layout. A concept may be, for example, represented as a document section or subsection. Other well-known examples include annotations and warnings. They may be presented on paper as boxes, footnotes, or different fonts; or presented electronically as buttons or separate windows. Any “highly structured” document seems to have clearly identifiable contents fragments: well-known examples include a contract for buying a house [Kilov, Ross 94], legal documents studied by the science of document analysis known as diplomastics [Duranti 91], or a relational database table. Various indexes\(^7\) obviously serve the same purpose. A document contents model may, however, be presented in a more explicit manner. For example, [Hehner 93] uses explicitly specified associations (close to Reference and Composition associations) in a quick tour of his book. For another example, [Bearman 93] refers to explicitly articulated requirements for record-keeping systems that include “a data model of relations between elements of information within a record”.

**AN EXAMPLE: MODELING TRADITIONAL LEGAL DISCLAIMERS**

Let us see how the ideas described above have been successfully used to understand, make more consistent, and use the collection of Bellcore legal disclaimers.

Bellcore disclaimers are quite complicated – a result of substantial work

\(^7\)Traditionally, a good index for a document may have been considered as a reasonable approximation of several concept maps representing several different viewpoints for a document: an index entry is a concept name!
by its legal organization. There exist numerous kinds of disclaimers (several dozen) for different kinds of documents. Each disclaimer consists of about a dozen components. These disclaimers are not fixed due to various reasons; new kinds of disclaimers may be needed for new kinds of documents. Document authors and users need to understand what material is included in disclaimers and which disclaimer to use in what circumstances. Bellcore legal organization needs to create new and update existing disclaimers, without starting from scratch each and every time. Moreover, consistent disclaimer templates should be provided to different users working on different document creation platforms. In other words, both users and legal organization need a reasonable way to manage the intellectual contents of disclaimers, using very different representation and possibly layout viewpoints.

As a result of document analysis, an information model of disclaimers as composite documents has been created and accepted by the legal organization. The information modeling approach successfully exposed non-trivial structure and relationships between different kinds of disclaimers and their components. Obviously, a model of an isolated disclaimer is quite simple, and does not provide substantial help in managing a collection of disclaimers. Only after understanding and specifying (modeling) different disclaimers was it possible to find out that they have a substantial amount of common contents components. These components have not always been presented as logical layout elements (such as paragraphs) because different attorneys – disclaimer authors – did not have this as an explicit goal. In some cases, the same contents elements were represented in different disclaimers using slightly different wordings, so that document analysis could not be based only on keywords and key expressions.

The disclaimer information model is clear and simple, providing both document authors and the disclaimer authors with an understandable succinct overview. The number of different (reusable and sometimes parameterizable, with a disclaimer name or its component name as an actual parameter) components of disclaimers has been substantially reduced, leading to a much better understanding of disclaimer contents. The total size of all currently existing disclaimers exceeded 60 pages making their understanding without an explicit model (abstraction) quite difficult. An explicit information model of the disclaimer collection made possible to create a collection of unique components, the total size of which was only about 6 pages. This happens because the same component often participates in more than one disclaimer type. Information modeling led to some improvements in certain disclaimer texts by making them more mutually consistent. New disclaimers may be created by attorneys from existing
components, using the "plug and play" approach. (Obviously, new components may be created if need arises.) Moreover, this approach led to easier maintenance of disclaimer texts and to a potential increased productivity of the legal department.

The top-level view of the disclaimer information model is presented below in a graphical form. As earlier, each association type (e.g., "Composition - ordered" (CO) or "Subtype - exhaustive" (SE)) is precisely defined using appropriate invariants [Kilov, Ross 94].

A more abstract model for disclaimers is possible: consider every disclaimer as a Composition of a "Header" and a "Body", with different subtypes of these components for different disclaimer subtypes. In this model, the concrete subtype-specific names of different components will not be specified. Observe, however, that qualified names like "TAudit. Header" or "General. Header" in the more abstract model below are equivalent to subtype-specific names like "Parties and Distribution" or "Description (and Review)" in the more concrete one above.

The components shown here are not elementary and therefore will themselves be represented as Compositions. The model of one of these components - a body of the disclaimer for a general document - may be presented (less graphically) as:

General document - body (a safety paragraph or empty)
revision (long)
Disclaimer

(a safety paragraph or empty)
warranty on information
non-suggestion (SR)
Bellcore Client Company requirements
license
non-recommendation.

Parameters are shown in parentheses. “A safety paragraph or empty” is
a formal parameter that will be instantiated and replaced either with the
actual parameter “a safety paragraph” or with the actual parameter “empty”,
leading to two different kinds of a body of the disclaimer for a general
document – with or without the safety paragraph. “Long” and “SR” are
actual parameters: the components “revision” and “non-suggestion” with
the same or different actual parameters are used in other disclaimers.

The disclaimers as composite fixed documents (for users) and as Com­
positions of reusable components (for disclaimer authors – the legal organi­
zation) may be stored only once, on paper or electronically. In particular,
the disclaimers may be stored in the World Wide Web, with a disclaimer
home page presenting the disclaimer model. This model makes the
disclaimers more understandable and therefore will save time and effort
(and money) of both document and disclaimer authors. As a result, the
disclaimer collection becomes more understandable and therefore better,
more consistent, and easily extensible. For disclaimer authors, the model

*This elementary component, for example, is encountered in 4 Compositions, references
to which are provided in the model used by the disclaimer authors:

Non-recommendation: Bellcore does not recommend, approve, certify, warrant,
guaranty or endorse products, processes or services and nothing contained herein
is intended or should be understood as any such recommendation, approval,
certification, warranty, guaranty or endorsement to anyone.
implies the set of operations available to create or update the text of a disclaimer. For disclaimer users, the model helps to obtain identical disclaimer texts, independent of the platform (either “standard” or non-standard) used by a document author. Thus, the information model provides a solid foundation for document understanding and document management reengineering.

Obviously, the same approach may be used for any document collection.

Conclusions

We don’t need to start from scratch in document management. As we have seen, generic information modeling concepts are perfectly reusable. This permits, in particular, use of the same familiar constructs, such as Composition, Subtyping, Reference, and so on, to understand and specify both an enterprise and a document that describes the enterprise. The mental structure of the information in the user’s head is the same for the application and for its documents. Such an approach clearly shows how to achieve the objective “to match the structure of the technical documents to the structure of the system being represented” [Paquel 94]. Software development may also be considered as a document-based process [Welsh 94] which uses both traditional documents (e.g., requirements) and structured ones (e.g., databases or program code); with the need to specify, clearly and explicitly, relevant fragments of these documents and relationships between these fragments.

The approach to document modeling shown here promotes understanding. It shows both the structural and behavioral properties of a document, a document fragment, or a document collection – at an appropriate granularity level for contents – without dealing with software (and hardware) tools used for implementing these properties. A document specification created in this manner is abstract and precise. It is understandable to and usable by domain experts, modelers, developers, and users. When the underlying concepts are simple, the customer will be able to find out whether the specification is complete and consistent – whether it correctly represents the requirements.

Acknowledgments

Thanks for useful discussions go to Pamela Keen, Virginia Meade and James Porter.
REFERENCES


