Appendix F – Part 9

Defendants' Supplemental Prior Art Statement
'228 Patent
(TC1471-TC1499)

to

TimeBase's Memorandum in Support of Its Motion for Summary Judgment of No Invalidity

Prior Art Analysis

experiments – data which is used repeatedly and updated as new information becomes available," at 654.

- *See, e.g.*, "The second activity of database management is to update, which includes the original storage of data, its repeated modification as things change, and ultimately, its deletion from the system when the data is no longer needed," at 655.

- *See, e.g.*, "The hiring of a new employee would cause a new record to be stored. Reducing available stock would cause an inventory record to be modified. Cancelling an airline reservation would cause a record to be deleted. All of these are recorded and updated in anticipation of future inquiries," at 655.

- **Bentley 1979:**

Bentley 1979 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Bentley 1979 discloses storing records which are larger than a single word. Bentley 1979 also does not require that these records be stored as entire documents. For example:

- *See, e.g.*, "The study of data structure for facilitating rapid searching is a fascinating subject of both practical and theoretical interest," at 397.

- *See, e.g.*, "In database terminology a file is a collection of records, each containing several attributes or keys," at 397.

- *See, e.g.*, "In a geographic database of U.S. cities…," at 398.

- *See, e.g.*, "To compile an honor list of older students…," at 398.

- **Campbell 1988:**

Campbell 1988 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Campbell discloses storing individual pages of a user manual. For example:

- *See, e.g.*, "A node contains arbitrary data that can be stored as text or as fixed-length binary blocks," at 856.

- *See, e.g.*, "UNIX manual pages provide a convenient example of how the HAM can model Intermedia webs. The manual page for the mail command is used to create a small web of information. Each document (manual page) is represented as a HAM node," at 859-860.

- **Fay 1996:**

Prior Art Analysis

Fay 1996 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Fay 1996 discloses that documents are broken down into subparts. For example:

- *See, e.g.,* "Each element of a document is associated with a data field. For example, one node of a tree may be a chapter, and contain textual data in the form of a chapter heading, a chapter introductory paragraph, a chapter abstract, etc., as well as "structural" data such as the identity of a parent (document), identity of children (sections), and connections to other places in other documents where the same language may also be used," at 1:25-32.

- **Horne 1997:**

Horne 1997 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Horne 1997 discloses storing statutes which are larger than a single word. Horne 1997 also does not require that these records be stored as entire documents. For example:

- *See, e.g.*, "Since the mid 1980s every statute and statutory instrument has been coded using SGML (the Standard Generalized Markup Language)," at 2.

- *See, e.g.*, "HMSO have a program called 'the Statute law Database'. This is an electronic version of Statutes in Force. It contains in SGML form the law as it was on a particular date in the 1980s together with all acts and statutory instruments which have come into force since that time," at 3.

- **Kim 1996:**

Kim 1996 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Kim 1996 discloses storing a document as several "atomic nodes." For example:

- *See, e.g.*, "Fourthly, since a composite object can refer elements, i.e., atomic objects, the composite object provides a syntax for the synchronization of its reference elements," at 497.

- *See, e.g.*, "The paradigm also provides some capabilities for supporting structured complex operations. Therefore, the object-oriented paradigm most closely represents the complex hypermedia data model," at 498.

- **Larson 1988:**

Larson 1988 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Larson 1988 discloses storing portions of text-based data. For example:

Prior Art Analysis

- *See, e.g.*, "A hypertext system is made up of a set of 'nodes' and 'links'. Nodes represent information sources in digital form. They may be segments of text . . . .," at 195.

- **Lo 1996:**

Lo 1996 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Lo 1996 discloses a system for storing, managing, querying, and displaying portions of data. For example:

- *See, e.g.,* "Hypertext is characterized by the notion of non-linear organization and presentation of textual information. The non-linearity is achieved by breaking down documents into a number of pages, which are then linked to each other in a network" at 23.

- *See, e.g.,* "Since structure can be recognized in documents, components such as chapters or sections can be recognized and hence be manipulated individually. The fragmentation model makes use of this advantage brought by the described structure to store only the modified components of a document when a new version of it is created," at page 32, text following heading "The Fragmentation Model."

- *See, e.g.,* "Möller recognizes that structured documents can be partitioned into a number of smaller units (fragments), each of which may contain one specific information item," at 34.

- *See, e.g.* section 2.3.4, starting on page 44, entitled "SGML Document Fragmentation."

- **Osterbye 1992:**

Osterbye 1992 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Osterbye 1992 discloses storing portions of documents that are greater than a single word, such as paragraphs or chapters. For example:

- *See, e.g.*, "Is it desirable to keep versions of the individual nodes and links, or do we only want to track entire 'hyperdocuments'?" at 33.

- *See, e.g.*, "Nodes are entities which have contents, and are specialized into atomic nodes which do not contain other entities, and composites which do contain other entities," at 34.

- *See, e.g.*, "In software engineering there are two levels of versioning. The lowest levels are the different modules that make up the programs. Each module can exist in several versions, and all the versions of a module is often referred to as a version

Prior Art Analysis

group," at 34.

- *See, e.g.*, "Nodes correspond to modules; notes will normally be short, e.g. sections or paragraphs rather than chapters, or routines rather than files," at 35.

- *See, e.g.*, "The node is the basic entity for storing contents. We require all nodes to have an attribute for contents and a name. Nodes are versioned," at 38.

- **Peltonen 1993:**

Peltonen 1993 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Peltonen 1993 discloses storing portions of text-based data. For example:

- *See, e.g.*, "*Documents* represent any data for design tools: drawings, manuals, bitmap images, etc. Originally we only used the concepts of a document and document version. However, our industrial partner turned out to require a considerably richer document structure. A document therefore includes a number of subdocuments, each subdocument has a number of subdocument versions, and each subdocument version has a number of representations," at 6.

- *See, e.g.*, Figure 2.

- *See, e.g.*, "Documents are composed of *subdocuments*. For instance, suppose a drawing comprises several sheets, each of which is manipulated as a separate file by the drawing tool. The sheets of the drawing as a whole form a single document, and each sheet is a subdocument. Subdocuments are also needed for a text document which includes figures made with a separate drawing program. The text file and the graphics files are stored in the database as separate subdocuments," at 7.

- **Povilus 1995:**

Povilus 1995 discloses, teaches or renders obvious this claim for the reasons stated by the Patent Examiner in the Office Action of March 24, 2000 (incorporated herein by reference) and others.

- **Promenschenkel 1995:**

Promenshenkel 1995 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Promenshenkel 1995 discloses storing sections of documents which can be compiled. For example:

- *See, e.g.*, "This allows components to be stored, manipulated and eventually assembled automatically as a magazine, electronic journal, book or in virtually any

Prior Art Analysis

other form chosen by the publishers," at 1.

- **Sacks-Davis 1994:**

  Sacks-Davis 1994 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Sacks-Davis 1994 discloses storing sections of text less than a full document. For example:

  - *See, e.g.*, "The last possibility we consider is to use an element-based model designed specifically for SGML documents. While alternatives that rely on retrieving multiple tuples or objects must then combine the objects to give the full text of the element, and those retrieving whole documents must decompose documents to extract the elements, the element approach supports retrieval by element directly," at THOM00198841.

  - *See, e.g.*, "It is also possible to index the atomic elements (objects or relations) and then define a join operation for text objects or relations that allows these results to be combined to access higher level elements, an approach applicable to either relational or object-based models," at THOM00198844.

- **Sciore 1994:**

  Sciore 1994 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Sciore 1994 discloses storing predefined portions of, for example, portions of employment records. For example:

  - *See, e.g.*, "Figure 3 illustrates an historical database in EXTRA-V. This scheme has three conceptual types: Person, Employee, and Company. A version of a conceptual object denotes a pervious or current state of the object. Each time a versioned attribute changes, a new version is created corresponding to the new state," at 87.

  - *See, e.g.*, "Each Employee version contains the information about an employee during some time interval," at 87.

- **Sacks-Davis 1995:**

  Sacks-Davis 1995 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Sacks-Davis 1995 discloses storing sections of text less than a full document. For example:

  - *See, e.g.*, "Rather than store documents as monolithic objects in a database it is more efficient to represent documents as a set of smaller fragments, which can be

Prior Art Analysis

connected by links," at 465.

- *See, e.g.*, "Once the structure is determined, a document can be partitioned into fragments to be stored in a database system," at 465.

- *See, e.g.*, "Since document parts, such as chapters or sections, are represented as subtrees within a document tree, the Hypertext table will typically contain an attribute identifying the parent of a node and another attribute identifying the sibling order of a node within a tree," at 465.

- **Travis & Waldt:**

  Travis & Waldt discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Travis & Waldt discloses that granularity can be chose by any particular user of an SGML database based on their particular needs. Travis & Waldt then goes on to recommend that granularity be selected at a level larger than a single word and smaller than an entire document. For example:

  - *See, e.g.*, "Currently, the documents are stored in either complete document form, or as very large document fragments. Access to smaller document fragments concurrently by multiple authors could expedite editing speed and allow for easier reuse of information. Storing document information as logical elements, which may be very small portions of documents, will create a large number of units of information that will need to be managed. Again, the DBMS is designed to manage large numbers of smaller information elements," at 17.

  - *See, e.g.*, "In implementing an SGML database, the most important factor is to determine the level of granularity. That is, which objects should be tracked and managed as a single piece. Sometimes this is a chapter or section, but it could be as large as the entire book or as small as every element or word," at 110-11.

  - *See, e.g.*, "The key to successful entity management is to break a document into logical pieces, which can be managed independently, while maintaining their identity in a particular document or set of documents," at 185.

  - *See, e.g.*, "The most important factor when implementing an SGML-enabled database is to determine the level at which the document information will be split. This is usually called 'granularity', and each piece is called a 'grain' or 'atom'." "Grains are typically chapters or parts, but we have seen implementations where each paragraph is stored separately." "Setting the grain size too large may cause unnecessary data access overhead." "Setting the grain size too small causes problems, also," at 202–03.

  - *See, e.g.*, "Determining the proper level of granularity is largely a matter of how the

Prior Art Analysis

data is structured, and what its purpose is in the enterprise," at 203.

- *See, e.g.*, "Sometimes, the level of granularity is self-defining. A common way is to break by chapter or sub-chapter. It is the level at which the author is likely to work. In many situations, even if a book has several authors or maintainers, a chapter will be owned by a single person. By setting the grain size to this object, the database reflects what the users do naturally," at 203.

- *See, e.g.*, "There are also technical factors to keep in mind when determining granularity. There is a certain amount of database overhead for each object. If each word were tracked, the overhead could easily be more than the actual data," at 203.

- *See, e.g.*, "A database loader in an SGML-enabled system contains some kind of "atomizer" that breaks the source document into the grain-sized pieces mentioned above," at 204.

- **Wilkinson 1998:**

Wilkinson 1998 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Wilkinson 1998 discloses a system for storing, managing, querying, and displaying portions of data. For example:

- *See, e.g.*, "[E]ither documents or their components have to be stored. The key task in using components is to decide on how documents are to be partitioned into components. The advantage in storing components instead of whole documents is that useful fragments can be retrieved without incurring the cost of retrieving whole documents," at 100.

- *See, e.g.*, "Alternatively, documents can be regarded as sets of fragments that have properties including type, parents, and children," at 103.

- **Wilson 1988:**

Wilson 1988 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Wilson 1988 discloses storing sections of legislation which are more than a word but less than a full document. For example:

- *See, e.g.*, "Justus automatically converts machine-readable versions of a variety of legal documents into hypertext documents for the Guide hypertext system," at 30.

- *See, e.g.*, "In a hypertext system, texts are divided into segments, sometimes called nodes," at 32.

- *See, e.g.*, "The text of a statute is particularly amendable to representation in a

Prior Art Analysis
hypertext system because it is already highly structured," at 32. • *See, e.g.*, "Each subsection label provided by the legal draftsmen is automatically converted by Justus into a node name or, in Guide terms, a definition button," at 32. • **Wilson 1990:** Wilson 1990 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Wilson 1990 discloses storing sections of legislation which are more than a word but less than a full document. For example: • *See, e.g.*, "Justus automatically converts machine-readable versions of a variety of legal documents into hypertext documents for the Guide hypertext system," at 30. • *See, e.g.*, "In a hypertext system, texts are divided into segments, sometimes called nodes," at 32. • *See, e.g.*, "The text of a statute is particularly amendable to representation in a hypertext system because it is already highly structured," at 32. • *See, e.g.*, "Each subsection label provided by the legal draftsmen is automatically converted by Justus into a node name or, in Guide terms, a definition button," at 32. • **Wilson 1992:** Wilson 1992 discloses "each portion is a block of the text-based data, the block being larger than a single word and less than the entirety of the text-based data." Specifically, Wilson 1992 discloses storing predefined portions of legislative acts such as sections, which are referred to in the article as nodes. For example: • *See, e.g.*, "An Act of Parliament may be divided into parts, sections, subsections, and paragraphs; a schedule, into subschedules, paragraphs and subparagraphs. An act must have at least one subsection; a schedule at least one paragraph. Hence, the text is already divided into segments…the text segments are the basic units of information, or lowest level nodes, of the hypertext system," at 161. • *See, e.g.*, "The Industrial Relations Act itself is a node that consists of the general description of the Act, nine Part nodes, and eight Schedule nodes," at 162. • **The Pre-1997 Westlaw/Westmate System**: The Westlaw/Westmate system contained predefined portions of legislation, which are

Prior Art Analysis

larger than words, but smaller than all Acts, or Codes. *See, e.g.*:

- Database Guide, at doc no. 79858–59 ("United States Code Annotated"): "A document is an annotated or unannotated section of USCA."
- www.westlaw.com: any annual statutory database prior to 1998.
- AMPEX § 2
- Wren 1994, at109–11, 141–42 (discussing searching statutory sections and showing attributes within a statutory section).
- The Essential Guide 1996, at 136 (showing that the text-based data within WESTLAW is stored as statutory sections)

- **The Pre-1997 Premise System**:

  The Premise system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes. *See, e.g.*:

  - Premise Statutes (including numerous statutory sections).

- **The Astoria System (pre-1997):**

  The Astoria System could be used with any documents or portions of documents, including legislation or portions of legislation, such as a section or blocks which are larger than words, but smaller than all Acts, or Codes. Use of this system therefore constitutes a method wherein each portion is a block of the text-based data, the block being larger than a single work and less than the entirety of the text-based data. For example:

  - *See, e.g.,* Astoria 1997-1: "Astoria . . . is a powerful yet east-to-use document component management system that provides the information repository and management infrastructure needed to help organizations capture critical business knowledge and distribute it more efficiently," at THOM00211907.
  - *See, e.g.,* XSoft Premiers Astoria: "Astoria . . . [is] a software system that allows groups of people to more easily collaborate on, create and edit massive or complex documents. Astoria is for use with 'structured' documents, which typically run into the thousands of pages, contain a series of reusable components such as headings, tables, and lists, and require multiple revisions or updates over many years," at THOM00198650.
  - *See, e.g.,* Astoria 1997-1: "Astoria lets users navigate through the document depository and view documents down to the individual components that comprise

Prior Art Analysis

them." at THOM00211907.

- *See, e.g.,* XSoft Astoria: "Astoria deals with the concept of 'document components.' A document component is a piece that is designed to be maintained as a unit, whether this be at the volume or book level, or at some finer granular point, such as paragraph or list," at THOM00198652.

- *See, e.g.,* Astoria 1997-1: "Astoria can apply revision information to only the components that change during an editing session. Astoria detects and maintains revision history at the component level, not just at the document level. . . Astoria stores versioning information in an efficient format, and past versions are always available for republishing or for providing an audit trail," at THOM00211908.

- *See, e.g.,* XSoft: "Because of its sophisticated integration with SGML editors, Astoria maintains revision information on individual elements, and past versions are always available," at THOM00198648.

- **The EnAct System** (previously known as Themis)**:**

  The EnAct system stores portions of text-based data that are larger than a word and smaller than an entire document. *See, e.g.*:

  - Arnold-Moore 1997-2, at 177–78: "In the *Themis* system we have chosen to fragment documents at the section level . . . . By using SGML to store the Statutes, we can automate the process of fragmenting large documents and only present to the user the part of the document that the user requests."

  - http://web.archive.org/web/19990430002036/www.thelaw.tas.gov.au/background.html : "All legislation in the database is broken up into a number of fragments (i.e. one fragment per Section or Schedule)."

- **The SCALEplus System:**

  The SCALEplus system stores portions of text-based data that are larger than a word and smaller than an entire document. *See, e.g.*:

  - Kerr 2000: Page 11-13, ¶ 490: "The standard unit of retrieval for legislation is a section of an Act or a regulation in Regulations . . . and for caselaw is the entire case. Users are able to modify the searchable scope of these retrieved documents."

  - SCALEplus Secrets, at 2: "SCALEplus has lots of information that is huge, particularly legislation. SCALEplus data is formatted in HTML which is common to all World Wide Web applications but is ideally suited for one or a few pages—to view a document you have to wait for the browser to load it (often over a modem). Because of this the decision was made to turn each piece of legislation into a number of HTML files, each file being a section of that Legislation. When a results list is

Prior Art Analysis

returned from SCALEplus what you see are the HTML files that have been found that match your search. For Legislation this will be a section of an Act; for Caselaw an individual case."

- **The Documentum/Interleaf System:**

  The Documentum/Interleaf system stores portions of text-based data that are larger than a word and smaller than an entire document. *See, e.g.*:

  - Interleaf provides support for compound documents, wherein documents are broken into components. *See* Ovum Interleaf 1996, at 256–57.

  - Documentum provides support for compound documents, wherein documents are broken into components. *See* Ovum Documentum 1996, at 210–212 ("Similarly, Documentum can be integrated with SGML parsers for importing SGML documents: using the parser, a large SGML document is turned into a set of compound documents, based on the internal document structure defined by the Document Type Definition.").

- **The Core Materials on Legal Ethics System:**

  The Core Materials on Legal Ethics system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.

- **The Federal Rules of Civil Procedure System:**

  The Federal Rules of Civil Procedure system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.

- **The Law Desk NY System:**

  The Law Desk NY system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.

- **The Law Desk USCS System:**

  The Law Desk USCS system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.

- **The New Mexico Law System:**

  The New Mexico Law on Legal Ethics system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.

- **The NY Official Reports System:**

Prior Art Analysis
The NY Official Reports system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes. • **The NY CLS Beta System:** The NY CLS Beta system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes. • **The OnPoint System:** The OnPoint system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes. • **The Social Security Plus System:** The Social Security Plus system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes. • **The UCC System:** The UCC system contained predefined portions of legislation, which are larger than words, but smaller than all Acts, or Codes.
Claim 13: Subject to the Court's claim construction, and given Defendants' understanding of Plaintiff's incomplete contentions regarding the construction and application of the claims, the following references disclose, teach or render obvious the following elements of Claim 13:
(a) A method for electronically searching legislation encoded with a markup language, the method comprising: • **Arnold-Moore 1994:** Arnold-Moore 1994 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Arnold-Moore 1994 discloses storing text-based legislative data encoded with SGML. Arnold-Moore 1994 further discloses a method for filtering through the stored legislative material based on effective dates. For example: • *See, e.g.*, "We discuss a data model for the storage, retrieval and display of legislation in large database collections," at Abstract. • *See, e.g.*, "SGML can be used to solve a number of the failings of the hyperbase approach. The problem of presentation is addressed by storing Acts of a given jurisdiction in SGML format satisfying a particular DTD," at *xii*.

Prior Art Analysis

- *See, e.g.*, "Both of these references discuss prototype systems which utilize SGML for hypertext database systems with legal applications," at *xii*.

- *See, e.g.*, "Queries can use these indices to return a subset of the database which can then be filtered on time constraints," at *xvi*.

- *See, e.g.*, "A time index on the version skeleton can be maintained independently of the content index to filter versions on time," at *xix*.

- **Arnold-Moore 1994-2:**

  Arnold-Moore 1994-2 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Arnold-Moore 1994-2 discloses storing text-based legislative data encoded with SGML. Arnold-Moore 1994-2 further discloses a method for searching the stored legislative material either by keyword, by attribute, or by a combination. For example:

  - *See, e.g.*, "A new class of document databases is emerging. These databases consist of large structured documents. Examples include databases of government legislation, maintenance manuals for systems as complex as aircraft carriers, and encyclopedia, and the documentation associated with a large software engineering project," at THOM00196608.

  - *See, e.g.*, "A data model and query language for accessing structured documents expressed in SGML is presented," at THOM00196608.

  - *See, e.g.*, "The database should also allow for partial document retrieval. The whole of a government Act may be an inappropriate retrieval unit, if one is searching for a definition. There may be a number of relevant portions of a single document that are relevant, and yet the whole document may still be an inappropriate retrieval unit," at THOM00196608.

  - *See, e.g.*, "The language, Structured Generalized Query Language (SGQL), allows efficient access to the content, structure and attributes of documents at any level within their structure," at THOM00196608.

  - *See, e.g.*, "Queries might specify that certain attributes have particular values, but have concentrated on matching against the content of the document," at THOM00196608.

  - *See, e.g.*, "These databases will need to be searched by attribute," at THOM00196608.

  - *See, e.g.*, "We thus see that a database system to support databases of large structured documents need a query language that allows retrieval: by exact matching Boolean combinations of words and phrases; by ranking by similarity to a given text;

Prior Art Analysis

using hypertext links; by attribute," at THOM00106609.

- *See, e.g.*, "Standard Generalized Markup Language (SGML), now provides a grammar for describing document structure which is widely used for document exchange," at THOM00196609.
- *See, e.g.,* Representative Queries at THOM00196609-10.
- *See, e.g.*, "The relational model extended to support content queries can support a whole range of queries including mixed content and structure, pure structure, and attribute queries," at THOM00196610.
- *See, e.g.*, "In order to construct a conceptual model of the database system we consider the database to be a list of ELF's (ELements with Features) where an ELF is: a complete SGML element … a list of features associated with that element," at THOM00196611.

- **Arnold-Moore 1997:**

Arnold-Moore 1997 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Arnold-Moore 1997 discloses storing text-based legislative data encoded with SGML. Arnold-Moore 1997 further discloses a method for searching the stored legislative material by keywords and by attributes such as effective date. For example:

- *See, e.g.*, "The Themis system manages a library of legislation which is encoded in the Structured Generalized Markup Language (SGML)," at 58.
- *See, e.g.*, "The drafter can view any Act or search the whole database using Boolean or ranking queries at any time point for which a valid version is stored on the system," at 59.

- **Arnold-Moore 1997-2:**

Arnold-Moore 1997-2 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Arnold-Moore 1997-2 discloses storing text-based legislative documents encoded with SGML. Arnold-Moore 1997-2 further discloses a method for filtering legislative material based on effective dates. For example:

- *See, e.g.*, "Themis uses SGML to store legislation," at 175.
- *See, e.g.*, "Despite improved performance of ranking (or natural language) queries, lawyers who have used digital legal libraries will be most familiar with the Boolean query approach," at 177.

Prior Art Analysis

- *See, e.g.*, "The SIM system supports both Boolean (with proximity operators) and ranking queries," at 177.
- *See, e.g.*, "Law librarians have identified the need for a standard coding of structure in legal texts for ease of distribution and reformatting, particularly identifying SGML as an appropriate scheme," at 177.
- *See, e.g.*, "Logical structure is identified by tags which appear interspersed with the text in an SGML document," at 177.
- *See, e.g.*, "A standard inverted-file word index can be used to extract a list of matching documents on content (or an enhanced index to allow queries on structure also), and then this list can be filtered by a time constraint using a time-index," at 179.
- *See, e.g.*, "Each fragment or table of contents has a valid start and end time associated with it. We then use an inverted-file index to give access to the fragments by content. As described for whole documents, we then filter results using the time information to collect just the fragments which are valid at the specified time defaulting to the current date (See Figure 1)," at 179.

- **Horne 1997:**

Horne 1997 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Horne 1997 discloses storing text-based legislative data encoded with SGML. Horne 1997 further discloses a method for searching the stored legislative data by attributes such as effective date. For example:

- *See, e.g.*, "Since the mid 1980s every statute and statutory instrument has been coded using SGML (the Standard Generalized Markup Language)," at 2.
- *See, e.g.*, "But markup could go further. It could give the dates on which the amendments were made, the dates on which they took effect, and the names of the Acts or SI which had made them, and the user's program could use this markup to display a statute as it was on a particular date chosen by the user and could offer hypertext cross-references to the amending legislation," at 3.
- *See, e.g.*, "HMSO have a program called 'the Statute law Database'. This is an electronic version of Statutes in Force. It contains in SGML form the law as it was on a particular date in the 1980s together with all acts and statutory instruments which have come into force since that time," at 3.
- *See, e.g.*, "There is a 'front-end' program which can be used to search the database and to display the results in HTML. So the user should be able to use the program to discover easily what the law is on a particular point on a particular date," at 3.

Prior Art Analysis

- **Lo 1996:**

  Lo 1996 discloses "A method for electronically searching legislation encoded with a markup language, the method comprising." Specifically, Lo 1996 discloses searching for and presenting text-based legislative documents. For example:

  - *See, e.g.,* "Examples of huge documents are an encyclopedia and the Acts of Parliament," page 11, section 1.2.3.
  - *See* page 27, under the heading "Types of Documents": "In the example of legal databases, the role of auxiliary documents is played by Amendment Acts . . . ."
  - *See, e.g.*, "SGML (Standard Generalised Markup Language) was adopted by ISO as an international standard to describe the structure of electronic documents. The reason for using SGML is its international acceptance as an electronic document markup standard. Furthermore, while the description of a document's structure is primarily applied in publication, database technology could also make use of this structural knowledge to enhance its management of documents," at 339.
  - *See, e.g.*, "SGML tags are placed in text to denote its structure and such practice is known as descriptive markup. A component in the text such as a title or a paragraph can be explicitly defined by marking them with these tags," at 339.
  - *See generally* pages 36–37.
  - *See, e.g.,* page 113: "All the attributes in both database are indexed by the SIM DBS and are thus searchable."

- **Promenschenkel 1995:**

  Promenschenkel 1995 discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Promenschenkel 1995 discloses storing a variety of text-based documents encoded with SGML. Promenschenkel 1995 further discloses a method for searching the stored text-based data. For example:

  - *See, e.g.*, "It will encompass capture and conversion of the article, Standard Generalized Markup Language (SGML) editing," at 1.
  - *See, e.g.*, "IDI's storage manager built to accept, query, retrieve and manipulate SGML document components as separate objects," at 1.
  - *See, e.g.*, "Because the system is set up uniquely for each individual organization, it can be used in a wide variety of industries to suit specific publication needs," at 2.

- **Travis & Waldt:**

Prior Art Analysis
Travis & Waldt discloses "a method for electronically searching text-based data encoded with a mark-up language." Specifically, Travis & Waldt discloses storing text-based data, such as legislation, encoded with SGML. Travis & Waldt further discloses a method for searching the stored text-based data either by keyword or by attribute, or both. For example: • *Passim*, especially Chapter 10. • *See, e.g.*, "SGML browsers offer context-sensitive searching capabilities so that the user can quickly access the required information . . . . For example, a search can be defined to allow a user to search for a part number, but only if it is contained in a chapter that was updated after a certain date. Or, a user can have the browser return a list of all sections containing a particular phrase, but only if the phrase is contained in a note. These are examples of context-sensitive searches," at 52-53. • *See, e.g.*, pages 194–95 (and figure 61), 198 (and figure 64). • *See, e.g.*, "The loader also makes available to the database parameterized information that can be used later to search and retrieve the appropriate objects. Such parameter information is object identifiers, author names, creation and modification dates, and perhaps some keywords. Most of this information can be obtained by querying the attributes on the element tags in the content of the document object," at 204. • *See, e.g.*, "Consider the requirement to create a link to an on-line database containing legal citations. The name of the citation must be rendered on the screen in a different color and underlined, which informs the user that the item is associated with an external link. . . . Notice the unique number of the citation is contained in the "num" attribute. This will be used to access the database, while the actual name of the citation is stated separately," at 306–07. • *See, e.g.* , Case Study: RIA TIGRE System, at 371-384. • **Wilkinson 1998:** Wilkinson 1998 discloses "a method for electronically searching legislation encoded with a markup language." Specifically, Wilkinson 1998 discloses a case study about a document management system for legislation (EnAct) which has documents encoded in SGML. For example: • *See, e.g.* Chapter 9, starting on page 161 (entitled "Case Study: Managing Legislation"). • **The Pre-1997 Westlaw/Westmate System**: The Westlaw/Westmate system discloses searching legislation encoded with markup

Prior Art Analysis

language, and so using the system would entail performing this method. *See, e.g.*:

- DataBasics 1993, ("United States Code Annotated"): Disclosing mechanisms for searching, at doc no. 79858-59.
- Westlaw DB 1991, (disclosing the mechanism for searching the Westlaw databases), at 2-5.
- *See generally* Wren 1994
- AMPEX § 2
- The Essential Guide 1996, at 3: "You can use WESTLAW to retrieve information from primary sources, such as cases and statutes from all 50 states and the District of Columbia, and from secondary sources, such as law reviews and treatises. You can seamlessly access Dow Jones News/Retrieval sources, including *The Wall Street Journal*, the same-day *New York Times* News Service and over 2,000 other sources. In addition, WESTLAW contains hundreds of databases from DIALOG, the world's largest online source of factual information. Subjects covered include business, current events, intellectual property, medicine, science and technology, and much more."

- **The Pre-1997 Premise System**:

The Premise system discloses searching legislation encoded with markup language, and so using the system would entail performing this method. *See, e.g.*:

- Premise Software & Statutes: Select "Search/Search Book…" and then change the "Search Using" field to "Fields Template," which will then list several of the attributes of the portions and amended portions.
- Premise Publisher: 30 (stating that the chapter is instructing the user how to insert markup code into documents for purposes of publishing them via PREMISE); 74–96 (explaining how to add the correct markup tags and attributes to an ASCII file); 359 (defining "Data markup"); 360 (defining "Data markup codes")
- Premise Research: Chapter 7 *generally* ("Retrieving Documents Using Descriptive Words")

- **The Astoria System (pre-1997):**

The Astoria System was used to search documents encoded with a markup language. For example:

- *See, e.g.*, Astoria 1997-1: "Astoria lets users navigate through the document depository and view documents down to the individual components that comprise

Prior Art Analysis
them." at THOM00211907. • *See, e.g.,* XSoft Astoria: "Astoria deals with the concept of 'document components.' A document component is a piece that is designed to be maintained as a unit, whether this be at the volume or book level, or at some finer granular point, such as paragraph or list." (THOM00198652) • *See, e.g.,* Astoria 1997-1: "Astoria provides a multilingual engine that lets users search on document content, structure, attributes, and version information," at THOM00211909. • *See, e.g.,* Screen shot, at THOM00211908. • *See, e.g.,* XSoft Astoria: "Astoria has complete version control capabilities. . . . The result is a version control module that is consistent with an SGML environment," at THOM00198652. • *See, e.g.,* XSoft Premiers Astoria: "Astoria is fully integrated with the XSoft InContext SGML Editor," at THOM00211913. "Astoria is particularly well-suited to working with structured documents based on the Standardized Generalized Markup Language (SGML), making the management of those documents significantly easier and more productive," at THOM00211914. "[I]t builds on the value of SGML with document management capabilities such as version control, revision tracking and component re-use," at THOM00211914. • *See, e.g.,* XSoft: "Astoria is an object-oriented document production component management system that enable users to easily find, use, share and manage SGML documents and their components, as well as unstructured documents. . . .Because Astoria works directly with SGML elements using an object-oriented database, it can provide unprecedented control over SGML documents as well as unstructured information by allowing fine-grained access and version control," THOM00198647. • **The EnAct System** (previously known as Themis)**:** The EnAct system discloses searching legislation encoded with markup language, and so using the system would entail performing this method. *See, e.g.*: • Arnold-Moore 1997-2, at 178, figure 1 (showing ability to search the text-based information) • http://web.archive.org/web/19990430002036/www.thelaw.tas.gov.au/background.html: "Legislation is stored in a format known as SGML." • **The SCALEplus System:**

Prior Art Analysis

The SCALEplus system discloses searching legislation, and so using the system would entail performing this method. *See, e.g.*:

- Kerr 2000: *Generally* Chapters 6 & 11.
- SCALEplus Secrets, at 2: "SCALEplus has lots of information that is huge, particularly legislation. SCALEplus data is formatted in HTML which is common to all World Wide Web applications but is ideally suited for one or a few pages—to view a document you have to wait for the browser to load it (often over a modem). Because of this the decision was made to turn each piece of legislation into a number of HTML files, each file being a section of that Legislation. When a results list is returned from SCALEplus what you see are the HTML files that have been found that match your search. For Legislation this will be a section of an Act; for Caselaw an individual case."
- SCALEplus UM 2: "SCALEplus presents all Law Databases obtained and/or prepared by Federal Attorney General's Department as Searchable and Browseable data." (THOM00221675)
- SCALEplus UM 2: "Search Results Page" screen shot, and the text describing this screen shot. (THOM00221679)
- SCALEplus UM 2: "Advanced Search Screen" at THOM00221692 and text describing the features on that screen, including the "Date Search Options."

- **The Documentum/Interleaf System:**

  The Documentum/Interleaf System discloses a method for electronically searching legislation encoded with a markup language. For example:

  - *See, e.g.*, Ovum Interleaf 1996, at 262 (and figure H2.6) ("It is possible to search on attribute combinations and combine these using Boolean operators.")
  - *See, e.g.*, Ovum Documentum 1996, at 220 (and figure H1.10) ("Documentum exposes just about all a document's attributes to querying.")
  - *See, e.g.*, Interleaf has been used to store legislation. *See* Consleg 1996, at 301 ("SGML is used as the representation format for the storage of acts.")

- **The Core Materials on Legal Ethics System:**

  The Core Materials on Legal Ethics system involves a method of using a system which has a means for searching its text-based information encoded with markup language.

- **The Federal Rules of Civil Procedure System:**

Prior Art Analysis
The Federal Rules of Civil Procedure system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The Law Desk NY System:** The Law Desk NY system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The Law Desk USCS System:** The Law Desk USCS system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The New Mexico Law System:** The New Mexico Law on Legal Ethics system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The NY Official Reports System:** The NY Official Reports system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The NY CLS Beta System:** The NY CLS Beta system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The OnPoint System:** The OnPoint system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The Social Security Plus System:** The Social Security Plus system a method of using a system which has a means for searching its text-based information encoded with markup language. • **The UCC System:** The UCC system a method of using a system which has a means for searching its text-based information encoded with markup language
(b) allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request;

Prior Art Analysis

- **Arnold-Moore 1994:**

  Arnold-Moore 1994 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Arnold-Moore 1994 discloses allowing a user to search while filtering search results based on time. For example:

  - *See, e.g.*, "Queries can use these indices to return a subset of the database which can then be filtered on time constraints," at *xvi*.
  - *See, e.g.*, "A time index on the version skeleton can be maintained independently of the content index to filter versions on time," at *xix*.

- **Arnold-Moore 1994-2:**

  Arnold-Moore 1994-2 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Arnold-Moore 1994-2 discloses allowing a user to search based on both content and attributes, such as date. For example:

  - *See, e.g.*, "A data model and query language for accessing structured documents expressed in SGML is presented," at THOM00196608.
  - *See, e.g.*, "The language, Structured Generalized Query Language (SGQL), allows efficient access to the content, structure and attributes of documents at any level within their structure," at THOM00196608.
  - *See, e.g.*, "Queries might specify that certain attributes have particular values, but have concentrated on matching against the content of the document," at THOM00196608.
  - *See, e.g.*, "These databases will need to be searched by attribute," at THOM00196608.
  - *See, e.g.*, "We thus see that a database system to support databases of large structured documents need a query language that allows retrieval: by exact matching Boolean combinations of words and phrases; by ranking by similarity to a given text; using hypertext links; by attribute," at THOM00106609.
  - *See, e.g.*, "Standard Generalized Markup Language (SGML), now provides a grammar for describing document structure which is widely used for document exchange," at THOM00196609.
  - *See, e.g.,* Representative Queries at THOM00196609-10.
  - *See, e.g.*, "The relational model extended to support content queries can support a

Prior Art Analysis

whole range of queries including mixed content and structure, pure structure, and attribute queries," at THOM00196610.

- *See, e.g.*, "In order to construct a conceptual model of the database system we consider the database to be a list of ELF's (ELements with Features) where an ELF is: a complete SGML element … a list of features associated with that element," at THOM00196611.

- **Arnold-Moore 1997:**

Arnold-Moore 1997 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Arnold-Moore 1997 discloses allowing a user to search on both content and date. For example:

- *See, e.g.*, "The drafter can view any Act or search the whole database using Boolean or ranking queries at any time point for which a valid version is stored on the system," at 59.

- **Arnold-Moore 1997-2:**

Arnold-Moore 1997-2 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Arnold-Moore 1997-2 discloses allowing a user to search both on content and date. For example:

- *See, e.g.*, "Despite improved performance of ranking (or natural language) queries, lawyers who have used digital legal libraries will be most familiar with the Boolean query approach," at 177.

- *See, e.g.*, "The SIM system supports both Boolean (with proximity operators) and ranking queries," at 177.

- *See, e.g.*, "Law librarians have identified the need for a standard coding of structure in legal texts for ease of distribution and reformatting, particularly identifying SGML as an appropriate scheme," at 177.

- *See, e.g.*, "Logical structure is identified by tags which appear interspersed with the text in an SGML document," at 177.

- *See, e.g.*, "A standard inverted-file word index can be used to extract a list of matching documents on content (or an enhanced index to allow queries on structure also), and then this list can be filtered by a time constraint using a time-index," at 179.

- *See, e.g.*, "Each fragment or table of contents has a valid start and end time

Prior Art Analysis

associated with it. We then use an inverted-file index to give access to the fragments by content. As described for whole documents, we then filter results using the time information to collect just the fragments which are valid at the specified time defaulting to the current date (See Figure 1)," at 179.

- **Bachman 1973:**

Bachman 1973 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Bachman 1973 discloses searching by attributes including date. For example:

- *See, e.g.*, "In sequential file technology, search techniques are well established. Start with the value of the primary data key, of the record of interest, and pass each record in the file through core memory until the desired record, or one with a higher key, is found. (A primary data key is a field within a record which makes that a record unique within the file.) Social security numbers, purchase order numbers, insurance policy numbers, bank account numbers are all primary data keys. Almost without exception, they are synthetic attributes specifically designed and created for the purpose of uniqueness. Natural attributes, e.g. names of people and places, dates, time, and quantities, are not assuredly unique and thus cannot be used," at 654.

- *See, e.g.*, "Database management has two main functions. First is the inquiry or retrieval activity that reaccesses previously stored data in order to determine the recorded status of some real world entity or relationship," at 654.

- *See, e.g.*, "In addition to a record's primary key, it is frequently desirable to be able to retrieve records on the basis of the value of some other fields. For example, it may be desirable, in planning ten-year awards, to select all the employee records with the 'year-of-hire' field value equal to 1964. Such access is retrieval by secondary data key. The actual number of records to be retrieved by a secondary key is unpredictable and may vary from zero to possibly include the entire file. By contract, a primary data key will retrieve a maximum of one record," at 655.

- *See, e.g.*, "With the advent of retrieval on secondary data keys, the previously one-dimensional data space received additional dimensions equal to the number of fields in the record," at 655.

- **Bentley 1979:**

Bentley 1979 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Bentley 1979 discloses six alternative methods for searching in the disclosed k-dimensional space using attributes. For example:

- *See, e.g.*, "In this section we investigate a number of search methods for range

Prior Art Analysis

searching," at 398.

- *See generally* 398-405.

- **Campbell 1988:**

Campbell 1988 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Campbell 1988 discloses searching the stored data by attributes, including date. For example:

- *See, e.g.*, "The HAM maintains history for these objects, allows selective access through a filtering mechanism…," at 856.

- *See, e.g.*, "Attributes are also used in the predicates that are part of the HAM filters," at 857.

- *See, e.g.*, "The HAM provides a filtering mechanism that allows subsets of HAM objects to be extracted from large graphs. Filters allow the user to specify visibility predicates, which are expressions relating to attributes and their values. HAM filters only return objects that satisfy the predicates. Filters also allow the user to specify a version time so that earlier versions of a graph can be examined," at 857.

- *See, e.g.*, "*Get operations* retrieve data from existing objects. A get operation takes an object index and a version time, and returns the data that existed at the specified time. The object index specifies a unique identifier for the object from which data is being retrieved. The version time is a time range for the data retrieval," at 858.

- *See, e.g.*, "*Filter* (and linearize) *operations* selectively retrieve information from a graph. A filter operation takes a predicate, a version time, and a list of attributes. These operations return a list of objects that satisfy the predicate and a list of requested attributes attached to each object. The version time specifies the time at which the filter is to search for the information. Each filter operation also has unique parameters in addition to those already specified," at 858.

- **Elmasri 1990:**

Elmasri 1990 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request ." Specifically Elmasri 1990 discloses searching information by ranges of effective dates, and therefore discloses this method. For example:

- *See, e.g.*, "A search for objects that satisfy such a temporal condition combines selection based on a time interval with a selection based on conditions involving attribute values." Summary of the Invention.

Prior Art Analysis

- *See, e.g.*, "Although the interval-based search problem is similar in many respects to the k-dimensional spatial search problem, the various methods proposed for the k-dimensional spatial search . . . are not suitable for the time dimension." Background of the Invention.

- *See, e.g.*, "A time index for temporal databases is provided which enables the retrieval of database object versions that are valid during specified time periods. Unlike prior access and retrieval structures, the present index is based on objects whose search values are time intervals rather than time points." Abstract.

- **Horne 1997:**

Horne 1997 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Horne 1997 discloses allowing a user to search on attributes, including date. For example:

- *See, e.g.*, "But markup could go further. It could give the dates on which the amendments were made, the dates on which they took effect, and the names of the Acts or SI which had made them, and the user's program could use this markup to display a statute as it was on a particular date chosen by the user and could offer hypertext cross-references to the amending legislation," at 3.

- *See, e.g.*, "There is a 'front-end' program which can be used to search the database and to display the results in HTML. So the user should be able to use the program to discover easily what the law is on a particular point on a particular date," at 3.

- **Kim 1996:**

Kim 1996 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Kim 1996 discloses allowing a user to search the stored text-based data based on attributes, such as date. For example:

- *See, e.g.*, "First, since our hypermedia markup language is designed using SGML, the language can ... support content-based and structure-based retrieval," at 496.

- *See, e.g.*, "Therefore, it is necessary to support content-based and structure-based retrieval as well as database mechanisms for hypermedia documents," at 496.

- *See, e.g.*, "Therefore, it is necessary to support an efficient information retrieval, which provides content and structure-based retrieval, and database query mechanism. Besides, the content-based retrieval method searches every object, which consists of the hypermedia network, and the object contents. On the other hand, the structure-based retrieval searches the logical and hypermedia network structures," at 498.

- *See, e.g.*, "Since the data about document structure and attribute values can be stored

Prior Art Analysis
as instances in the database, Postgres can directly process the structure-based retrieval. The other is a content-based retrieval. When a content-based retrieval query is given, the information retrieval manager performs full-text retrieval against the hypermedia document database," at 500. • *See, e.g.*, "For this reason, we design a new query language which supports both an information retrieval mechanism and a database query mechanism for handling structure hypermedia documents," at 500. • *See, e.g.*, "Element attribute search," at 501. • **Kimball 1996:** Kimball 1996 discloses "a plurality of attributes, each attribute being a point on an axis of a multidimensional space for organizing and displaying." and so using the system would entail performing this method. For example: • "Fundamentally, this is a book about dimensional modeling and how to build a dimensional data warehouse and keep it running. Dimensional modeling is a new name for an old technique for making databases simple and understandable. When a database can be visualized as a 'cube' of three, four, or even five or more dimensions, people can imagine slicing and dicing that cube along each of its dimensions." Page xx. • Text following the heading "The Time Dimension," Pages 7–8. • Text following the heading "The Dimensional Model," Pages 10–11. • Text following the heading "The Dimensional Tables," Pages 13–14. • Text following the heading "Attributes Are the Drivers of the Data Warehouse," Pages 17–18. • **Larson 1988:** Larson 1988 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Larson 1988 discloses attribute/value pairs that can be searched by a user. For example: • *See, e.g.*, "Neptune also permits any number of descriptive attribute/value pairs to be assigned to any node or link, which may be searched," at 196. • *See, e.g.*, "The indexes supported in Telesophy include a keyword index (providing Boolean and proximity searching), a 'temporal index' that permits selection by the time an IU was created, and a 'spatial index' that 'places items in an N-dimensional

Prior Art Analysis
space based on their attributes, then allows the space to be searched," at 197. • **Sacks-Davis 1994:** Sacks-Davis 1994 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Sacks-Davis 1994 discloses allowing a user to search the stored text-based documents on attributes, such as date. For example: • *See, e.g.*, "SGML's power can be utilized to create additional types of query over the whole database on the structural characteristics of the documents," at THOM00198839. • *See, e.g.*, "We also want to be able to query on SGML attributes, for instance: Query 7.1 Find <corres>s with attribute confidential = yes," at THOM00198839. • **Sacks-Davis 1995:** Sacks-Davis 1995 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Sacks-Davis 1995 discloses allowing a user to search the stored text-based documents on attributes, such as date. For example: • *See, e.g.*, "Text retrieval systems should have the ability to manipulate structured information and attributes, as do conventional database systems. Thus a text system should, for example, be able to refer to a paragraph within a section, a date of publication, or to one of a list of authors," at 454. • *See, e.g.*, "SGML is a widely used standard for the representation and interchange of documents. SGML defines a language that can be used to specify abstract grammars consisting of tags that are interspersed throughout the text of documents," at 464. • **Sciore 1991:** Sciore 1991 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Sciore 1991 discloses allowing a user to search based on attributes such as "occuredAt" which records the date of a change. For example: • *See, e.g.*, "At the conceptual level, we show how versions can be chosen from a version set based on the values of their attributes," at 356-57. • *See, e.g.*, "The attribute occuredAT records the time at which the change took place. This attribute can be thought of as defining a one-dimensional time line, and allows the version set to be viewed as a function from times to versions," at 364.

Prior Art Analysis

- *See, e.g.*, "In general, the attributes chosen as dimensions should form a key of the version set, so that at most one version is associated with any coordinate in the version space," at 366.

- *See, e.g.*, "Our framework provides the means by which a database designer can specify a multi-dimensional logical structure to the version set. This logical structure can then be used to choose versions easily and conveniently," at 367.

- **Sciore 1994:**

  Sciore 1994 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Sciore 1994 discloses allowing a user to search on multiple attributes, including date. For example:

  - *See generally* 81-83.

- **Stonebraker 1990:**

  Stonebraker 1990 discloses "allowing a user to select a version date as a primary attribute of a multidimensional space and to input at least one search request." Specifically, Stonebraker 1990 discloses that users may query the POSTGRES database using attributes, such as date. For example:

  - *See generally* section II.B, starting on page 127 (demonstrating numerous examples of queries using attributes).

- **Taylor 1994:**

  Taylor 1994 discloses "a plurality of attributes, each attribute being a point on an axis of a multidimensional space for organizing and displaying." Specifically, Taylor 1994 discloses the ability to search on the time dimension as well as other dimensions. For example:

  - "SHIC provides for multiple classifications of an item which is crucial for classifying media items such as photographs where one scene could be classified in several different ways," at 240.

  - "The temporal classification schema . . . stores information relating artefacts to some existence in time. The schema consists of year time point identifiers that constitute the relevant temporal span," at 240.

  - *See* figure 1.

  - "The combination of the conceptual, temporal and geographical classification spaces and their respective operators means that information requests do not need to be