IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

HYPERPHRASE TECHNOLOGIES, LLC and
HYPERPHRASE INC.,

Plaintiffs,

v.

GOOGLE INC.,

Defendant.

Civil Action No. 06 C 0199 S

TABLE OF CONTENTS FOR EXHIBITS ATTACHED TO THE SECOND DECLARATION OF W. BRUCE CROFT, PH.D.

EXHIBIT 1

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

HYPERPHRASE TECHNOLOGIES, LLC and
HYPERPHRASE INC.,

Plaintiffs,

v.

GOOGLE INC.,

Defendant.

Civil Action No. 06 C 0199 S

SUPPLEMENTAL EXPERT REPORT OF W. BRUCE CROFT, PH.D ON INVALIDITY

I. BACKGROUND

1. I am a Professor in the Department of Computer Science of the University of Massachusetts, in Amherst, Massachusetts. I am also the Director of the Center for Intelligent Information Retrieval. The Center for Intelligent Information Retrieval is also at the University of Massachusetts.

2. My current curriculum vitae is attached as Appendix A to this document.

3. I am being compensated for my work in this matter at my customary rate of $400 per hour for work up to trial, and at the rate of $500 per hour at trial and at deposition.

II. SCOPE OF WORK

4. Generally speaking, the scope of the work I have undertaken in connection with this supplemental report is the same as in connection with my original report. However, I understand that there have been several developments since my original report:

i. The only remaining asserted claims are claims 1 and 7 of United States Patent No. 5,903,889 (the "'889 patent"), and claims 1, 24, 27, and 86 of United States Patent No. 6,516,321 (the "'321 patent").

ii. The United States Supreme Court has issued a decision addressing the law of obviousness, set forth below.

iii. The District Court in this case issued a summary judgment ruling finding non-infringement.

iv. The Appeals Court in this case issued an opinion affirming the District Court in part, and reversing in part.

5. A list of the additional materials that I have considered in connection with this supplemental report is attached as Appendix B.

III. SUMMARY OF OPINIONS

6. My opinion regarding the skill level of the person of ordinary skill in the art has remained unchanged.

7. Each of claims 1 and 7 of the ’889 patent and claims 1, 24, 27, and 86 of the ’321 patent is invalid as anticipated, and as obvious, in view of the prior art, for the reasons stated in my original report, (except to the extent provided in this supplemental report) and in this supplemental report, applying as applicable (and in the following order of precedence) the Appeal Court claim constructions, the District Court claim constructions, and the claim constructions set out in my expert report on non-infringement, unless otherwise indicated. Claims 24 and 86 of the ’321 patent are also invalid as indefinite, for the reasons stated in my original report.

IV. APPLICABLE LEGAL STANDARDS

8. Although I am not an attorney, and will not offer opinions on the law, I am informed of several principals concerning patent invalidity, which I have employed in arriving at my conclusions in this report.

i. Each patent claim is considered separately.

ii. An issued patent is presumed valid.

iii. This presumption is overcome and a patent claim is invalid as "anticipated" if every feature of the claim is found, expressly or inherently, in a single reference that is prior art to the claim.

iv. This presumption is overcome and a patent claim is invalid as "obvious" if it would have been obvious to a person of ordinary skill in the art at the time the invention was made. I understand that, in undertaking this analysis, it is important to consider four issues: (1) the scope and content of the prior art; (2) the differences between the prior art and the claim under consideration; (3) the level of ordinary skill in the art; and, (4) any "secondary considerations" of non-obviousness. I understand this fourth category to include "long-felt need" for the patented invention, that is, whether the field had been searching unsuccessfully for a solution to the problem ultimately solved by the claimed invention. I understand other factors related to this fourth category include "commercial success" of the patented invention, and whether others had previously tried, but failed, to implement the claimed invention. I further understand that a combination of prior art elements according to known methods may be obvious when it does no more than yield predictable results. I also understand that if a technique has been used to improve one device, system, or method, and a person of ordinary skill in the art would have recognized that it would likewise improve similar devices, systems, or methods, then using the technique in similar devices would have been obvious unless doing so would have been beyond that person's skill. More generally, I understand that I can look to interrelated teachings of multiple prior art references, to demands in the design community or the marketplace, and to the background knowledge possessed by a person of ordinary skill in the art to provide a reason why it would have been obvious to a person of ordinary skill in the art to combine known prior art elements in the way a patent claims. One way, but not the only way, to provide such a reason is to find a teaching, motivation, or suggestion in the prior art, the nature of the problem to be solved, or the knowledge of a person of ordinary skill in the art to combine the prior art elements in the manner claimed.

V. COMPARISON OF PRIOR ART TO ASSERTED CLAIMS

9. A detailed comparison of the asserted claims against the relevant prior art is made in the Exhibits attached as Appendix C to this report, and to my original report. My evaluation has been made from the perspective of a person of ordinary skill in the art at the time of the patents, and how he or she would have interpreted the disclosures in each prior art reference.

10. I consider the prior art discussed in this report and in my original report to be more relevant than the prior art considered by the patent examiners during the prosecution of these two patents.

A. The Asserted Claims

1. '889 Patent, Claims 1 and 7:

11. Claim 1 reads as follows:

> ***Claim 1.*** *A computer system with a plurality of data records on a plurality of databases, and a standardized format for addressing said data records, said computer system comprising:*
> *(a) a user interface having an interactive display program for requesting one of said data records and displaying a plurality of interface supported data formats;*
> *(b) means for receiving a reference to a first data record from said interactive display program;*
> *(c) means for retrieving said first data record;*
> *(d) means for parsing said first data record to identify a reference to a second data record;*
> *(e) means for modifying said reference to said second data record to create an address, said address being operable to retrieve said second data record; and*
> *(f) means for sending said modified first data record to said interactive display program.*

12. Claim 7 depends from claim 1, and reads as follows:

> ***Claim 7.*** *The computer system of claim 1, wherein said reference to said second data record* ***comprises a keyword phrase.***

13. The only substantive difference between independent claim 1 and dependent claim 7 is that claim 7 specifies that the reference in the first data record to the second data reference comprises a keyword phrase.

14. I am informed that with regard to means-plus-function elements recited in a patent claim there must be a disclosure in the specification of the corresponding structure of the limitation, in such a manner that a person of ordinary skill in the art will know and understand what structure corresponds to the claim limitation and, furthermore, that a mere statement that known techniques or methods can be used does not satisfy the requirement to actually disclose corresponding structure.

15. With reference to the '889 patent, Dr. Thompson's earlier infringement charts identify various portions of the specification that he asserts correspond to the means-plus-function language found in the asserted claims. I note the following regarding these cited portions of the disclosure: (1) regarding the "means for receiving," the cited passage (Dr. Thomposon's '889 claim charts, claim 1, element (b)) does not fully enable this claim because it does not say how the means receives a reference to a first data record from the interactive display program; the cited passage instead concerns converting data from one format to another compatible format, and not "receiving;" (2) regarding the "means for retrieving," the cited passage (Dr. Thompson's '889 claim charts, claim 1, element (c)) concerns "receiving" not "retrieving," which are plainly different terms in the same claim and must have different meanings, so the cited support here also fails to enable the claim; and (3) regarding the "means for modifying," the infringement charts identify the so-called "HyperText Cipher" as the corresponding disclosure, which "converts any text portion of the selected data record into a browser compatible format, such as HTML format, and any graphics video, or other non-text information into a browser, plug-in, or Java compatible format." See col. 9, ll. 2-6. This passage (Dr. Thompson's '889 claim charts, claim 1, element (e)) contains virtually no enabling disclosure of what these graphics, audio, video, or other non-text formats are, or how they are converted from any number of standard and proprietary formats to other formats that are

somehow compatible with the identified items, which are also not disclosed in any meaningful detail. Essentially, "HyperText Cipher" is a black box. It is also unclear, in view of the claim limitations and the cited support, how the browser that generally performs the "user interface" and "interactive display program" features is to be altered to achieve the function or results claimed in the '889 patent. Thus in view of my analysis above, claim 1 of the '889 patent fails to adequately disclose the corresponding structure of the means-plus-function limitations and is invalid.

16. For a comparison of the disclosure of each prior art reference to the asserted claims on a limitation by limitation basis, please refer to the claim charts attached as Appendix C to my original report, and Appendix C of this supplemental report.

17. Thistlewaite/Pastime: While I will leave much of the detailed analysis of this reference to my claim charts and my original report, I will address several alleged points of distinction raised by Dr. Nirenburg. I disagree with Dr. Nirenburg that this reference does not disclose a "plurality of databases." It clearly does. Specifically, it discloses a hyperbase that is made up of several distinct "sub-collections," including Senate Hansards, Standing Orders, Biographies, etc. See for example page 171. Each sub-collection is a separate "database." Dr. Nirenburg's position is unsupported, and if accepted, would make it impossible to know whether a given collection of records is found in one "database" or plural "databases," thus rendering the term "database," and the claim, invalid as indefinite. The reference also discloses a standardized format for addressing the data records in the databases, as described under item 2 in section 5.1 of the reference, in the discussion of the document identifier index.

18. Dr. Nirenburg also asserts that this reference does not disclose a user interface that supports a plurality of data formats (element a). Dr. Nirenburg has apparently overlooked that the reference expressly discloses plural data formats at p. 170: "Parliament provides data in a number of formats, including a text database format, a word-processing format, and ASCII." To the extent that Dr. Nirernburg, in stating at page 92 of his report that the reference "only discloses text support, not, for example, images or multimedia," intends to imply that a "plurality

of interface supported data formats" must include non-text formats, that is a claim construction issue that I understand will be addressed by the Court. In any event, I do not see any basis in the claim for such a narrow interpretation of this term. Moreover, the "user interface having an interactive display program" referred to in element "a" concerns the capabilities of the client's system. This article expressly states that an HTML file is delivered to the client's browser software. HTML, and browser software, at the relevant time, were capable of "displaying a plurality of interface supported data formats," including text, graphics, and multimedia.

19. Dr. Nirenburg also asserts that this reference does not disclose modifying the reference, in the first data record, to the second data record (element e), and then sending the modified first data record to the display program. I disagree with this assertion as well. Dr. Nirenburg is apparently referring to the preferred embodiment discussed in the article. In that embodiment, in response to a request for a particular page, the computer program dynamically generates an HTML page. These dynamically generated HTML pages are cached (stored) on the server-side, and then the cached versions retrieved if subsequently requested. Thus, as to element f of the claim, the stored, modified first data record is sent to the client's browser. As to element e, whether this element requires that the underlying first data record (the original file) is itself modified, this too is a claim construction issue that I understand will be addressed by the Court.

20. In any event, even on Dr. Nirenburg's interpretation of element "e," this feature is clearly disclosed in the reference. The reference clearly states at p. 166 that the standard approach to create HTML documents is "to take the original documents, add HTML markers for formatting the documents presentationally, and add link information into documents which contain link sources. The next step is to store the resulting HTML documents in a file system, and make them visible to the Web via a HTTP server running on their system." While the reference goes on to explain why it is not necessary to store the modified HTML version of the document, it nonetheless clearly discloses this as an alternative approach. It is easy to see that there may be situations where it is desirable to keep the original document "clean," and others in

which it would be okay to modify the original document. For example, as to the former, if the original data were authored by another entity, and/or may be used by other users for different purposes, then it would probably be preferable not to modify the originals. Otherwise it might be beneficial to modify the underlying records, so that the system will be faster by virtue of not having to recreate the HTML documents every time they are requested. In short, this is a simple, and commonplace design choice. The article discloses both, and then for the preferred embodiment selects the option of not modifying the underlying record in view of the particular circumstances presented there. To the extent that the reference is determined not to disclose modification of the underlying data records, in view of the considerations outlined above, it certainly would have been obvious to do so.

2. ‘321 Patent, Claims 1 and 24:

21. Claim 1 reads as follows:

> ***Claim 1.*** *A method for identifying a referenced record referenced in a referencing record wherein the referenced record is referenced in the referencing record by at least a combination including a data reference (DR) and a modifier reference (MR), the method comprising the steps of:*
> *(i) receiving the referencing record;*
> *(ii) analyzing the referencing record to identify a DR, when a DR is identified:*
> *(a) identifying an MR rule set (MRRS) specifying the relationship between an MR and the DR;*
> *(b) analyzing the referencing record in accordance with the MRRS to identify the existence of the MR and, when the MR is identified;*
> *(c) identifying the referenced record associated with the DR/MR combination.*

22. Claim 24 depends from claim 1, and reads as follows:

> ***Claim 24.*** *The method of claim 1 further* ***including the step of linking the record reference to the referenced record.***

23. The only substantive difference between independent claim 1 and dependent claim 24 is that claim 24 further requires the step of linking “the record reference” to the referencing record. As noted in my original report, claim 1 does not define a “record reference,” and it is not at all clear whether this refers to (1) the DR, (2) the MR, (3) the DR/MR combination, or (4) something else. This lack of clarity renders claim 24 invalid as indefinite.

24. For a comparison of the disclosure of each prior art reference to the asserted claims on a limitation by limitation basis, please refer to the claim charts attached as Appendix C to my original report, and Appendix C of this supplemental report.

25. Thistlewaite/Pastime: While I will leave much of the detailed analysis of this reference to my claim charts and my original report, I will address several alleged points of distinction raised by Dr. Nirenburg. As to Dr. Nirenburg's suggestion that this reference does not disclose "an automated computer process that is 'on line' and inspecting the text of records," that is clearly incorrect. At page 171, the article plainly states that the HTML page is dynamically generated in response to a user request for the page. As to Dr. Nirenburg's alleged "two pass" distinction, he does not explain the basis for his position, nor does he say whether he agrees that the claim requires "two passes" through the document. This too is a claim construction issue that I understand will be addressed by the Court.

26. The remainder of Dr. Nirenburg's alleged points of distinction concern the issue of whether text such as "Prime Minister" can be considered a DR, and year information an MR. While Dr. Nirenburg apparently agrees that the former is a DR, he disputes that the year information is a MR, because he contends that the article does not say whether this year information is a term in the document, as opposed to being "metadata" in the document, but not in the text itself. Dr. Nirenburg does not define the what he means by "metadata." In fact, information such as date information can be present in an HTML document along with the text to be displayed and yet not be displayed itself. That date information is still a part of the document. In any event, and while this too is an issue of claim construction, there is nothing in the claim, or in the definition and usage of the term MR in the patent, that would preclude "metadata" from constituting an MR. If the reference were found to only disclose this date information in metadata, and if the claim were limited as Dr. Nirenburg suggests, then in any event it would be obvious to look to the document text to ascertain the relevant year information. The article expressly states at page 172 that the computer program could "take attribute information for the document (in this case, the date), which enables the identification of the correct target to be

computed." Thus, the reference is agnostic about the particular source of the information. The key is that it the article recognizes that terms such as "Prime Minister" can be ambiguous, and further recognizes that that ambiguity can be resolved by looking for further information in the document. On this particular issue of extracting date information from the document, as discussed in the next paragraph the article also expressly states that such information can be ascertained from the document text. On the issue of MRRSs, Dr. Nirenburg is also incorrect when he says at page 98 that "[n]othing in PasTime specifies relationships between the text 'Prime Minister' and a document date." The reference clearly discloses a rule, namely, when a member of Parliament is referenced, then extract date information in order to ensure that the correct biography is linked-to. I also disagree with Dr. Nirenburg's assertion at page 99 that "PasTime does not 'identify' any referenced record associated with a DR/MR combination … or 'associate' the DR/MR combination with a referenced record. … Instead, only the 'Prime Minister' text is 'identified' or 'associated' with a match to a target document while the document date is not, as it is derived from outside the document." Dr. Nirenburg is simply misreading the claim. The claim states that the referenced record associated with the DR/MR combination must be identified. In this example, the associated referenced record is the Prime Minister's biography for the year in question, and the article expressly discloses that it is identified. There is no requirement in the claim, as Dr. Nirenburg presumes in his critique, that the DR/MR combination is identified and associated with the target document, such as by making the combination the anchor of a hyperlink.

27. Moreover, Dr. Nirenburg also overlooks that there is another example of a DR/MR combination in this article, and that this second example also anticipates claim 1. The article explains at page 170 that the program can look for the keyword "bill". When it locates this keyword, it then applies a rule, namely, to look in the text for date information that will help determine which bill is being referenced. Thus, the keyword "bill" is the DR. When identified, a rule set (i.e., look for date information in the text) is used that specifies the relationship between an MR (the year) and this DR. The rule is then used to look for the information. When

the year is identified, the referenced record associated with this particular DR/MR combination (i.e., the bill from the appropriate year) is identified.

28. As to claim 24, as noted above it is not possible to determine what "the referenced record" is. However, this article discloses that the DR, the MR, and the DR/MR combination can all be used as the anchor text for a link, and so, if claim 24 is not invalid for indefiniteness, it too is invalid for anticipation.

29. While I will leave much of the detailed analysis of the other references to my claim charts and my original report, I note that, with regard to many of these other references Dr. Nirenburg's principal if not sole critique is that they allegedly do not disclose searching for an MR once a DR has been identified in a record. To the extent any of these references is found not to make such a disclosure, it would clearly have been obvious to have done so. The idea of looking for "modifier information" once one has encountered a term that is either ambiguous or indeterminate was commonplace long prior to these alleged inventions. It is also a notion that is by no means limited to the computer field. This is precisely what human beings have always done when reading a text. If one were to skim a newspaper article and see a reference to the last name shared by an old acquaintance, it is simple and commonplace to then look to the surrounding text for information such as first name, city of residence, age, etc. (or other such "rules") to see if the person being referenced is indeed the old friend. That claim 1 moves this normal human process into the computer realm is hardly non-obvious. Once one has the technology to find a certain set of keywords in a document (such as in Anthony, Justus, and van Hoff), that technology can easily be extended to provide for a further layer of detailed analysis to find other keywords that relate to, modify, or further define that keyword. Beyond common human experience, such an approach is disclosed in Thistlewaite, and also in Graham.

3. '321 Patent, Claim 27:

30. Claim 27 reads as follows:

> ***Claim 27.*** *A method to be used with a rule set including subject matter specific tag pairs and corresponding search rules, a separate tag pair for each of a plurality of different information types and a separate search rule for each pair, each pair including a begin tag and an end tag, the method comprising the steps*

> *of:*
>
> *(a) receiving a record;*
>
> *(b) examining the record according to the search rules to identify record segments including information of each of the information types;*
>
> *(c) when a record segment is identified which is of a particular information type: accessing the tag pair associated with the information type; inserting the begin tag before the identified segment and inserting the end tag after the identified segment.*

31. For a comparison of the disclosure of each prior art reference to the asserted claims on a limitation by limitation basis, please refer to the claim charts attached as Appendix C to my original report, and Appendix C of this supplemental report.

32. MUC-6 Named Entity Recognition: As stated in my original report, the automatic search for and tagging of entities in text with subject-matter specific begin and end tags was well known in the art for years before HyperPhrase filed its first patent application. Named entity recognition is one example of such tagging. MUC-6, the sixth in a series of Message Understanding Conferences, was held in November 1995. One of the goals of MUC-6 was to investigate and develop processes for accomplishing named entity recognition, specifically, the recognition of entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions. This task was intended to be of direct practical value (in annotating text so that it can be searched for names, places, dates, etc.) and an essential component of many language processing tasks, such as information extraction. The MUC-6 Named Entity Task Definition (Version 2.0, 31 May 95) included the following Task Overview (see, for example, http://cs.nyu.edu/faculty/grishman/NEtask20.book_3.html#HEADING4) (June 2, 1995):

2 TASK OVERVIEW

2.1 Markup Description

The output of the systems to be evaluated will be in the form of SGML text markup. The only insertions allowed during tagging are tags enclosed in angled brackets. No extra whitespace or carriage returns are to be inserted; otherwise, the offset count would change, which would adversely affect scoring.

The markup will have the following form:

```
<ELEMENT-NAME ATTR-NAME="ATTR-VALUE" ...>text-string</ELEMENT-
NAME>
```

Example:

```
<ENAMEX TYPE="ORGANIZATION">Taga Co.</ENAMEX>
```

The markup is defined in SGML Document Type Descriptions (DTDs), written for MUC-6 use by personnel at MITRE and maintained by personnel at NRaD. The DTDs enable annotators and system developers to use SGML validation tools to check the correctness of the SGML-tagged texts produced by the annotator or the system. The validation tools are available to MUC-6 participants in the file called muc6-sgml-tools. Annotators are using a software tool provided for MUC-6 by SRA Corporation to assist in generating the answer keys to be used for system training and testing.

2.2 Named Entities (ENAMEX tag element)

This subtask is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via the TYPE attribute as follows:

ORGANIZATION: named corporate, governmental, or other organizational entity

PERSON: named person or family

LOCATION: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)

2.3 Temporal Expressions (TIMEX tag element)

This subtask is for "absolute" temporal expressions only; explanation is provided in appendix B. The tagged tokens are categorized via the TYPE attribute as follows:

DATE: complete or partial date expression

TIME: complete or partial expression of time of day

2.4 Number Expressions (NUMEX tag element)

This subtask is for two useful types of numeric expressions, monetary expressions and percentages. The numbers may be expressed in either numeric or alphabetic form.

The task covers the complete expression, which is categorized via the TYPE attribute as follows:

MONEY: monetary expression

PERCENT: percentage

33. As set forth in the attached claim chart, this disclosure clearly anticipates claim 27. It discloses plural subject matter specific tags, such as PERSON, LOCATION, and DATE. These tags are used in pairs with beginning and end tags, such as <PERSON> and </PERSON>. A person of skill in the art would clearly have understood this overview to disclose a set of search rules that would allow the system to discriminate between different data types. The system would receive a record, and a computer program would examine the record according to the search rules to identify segments of the record that include information of each type. When such segments are located, the program would insert the tags for that information type around the segment.

34. While this overview was a statement of the objective of this aspect of MUC-6, in my opinion a person of skill in the art at this time could have implemented a system to achieve this objective without undue experimentation. For example, such a system could have been implemented using simple pattern matching (for example, to identify dates), and/or comparison of words in the record to a known database (for example, of person names and/or names of locations). In fact, numerous people were in fact able to develop such computer programs in just a few months, as reported in the proceedings for the conference, held in November, 1995. Two examples of such systems are discussed further below.

35. Aberdeen (Aberdeen, J. et. al., MITRE: Description of the Alembic System Used for MUC-6, Proc. of the 6th Conf. on Message Understanding, pp. 141-55, Columbia, Maryland (Nov. 6-8 1995)): This article describes one computer program, called Alembic, developed by several people at MITRE Corporation, to achieve Named Entity Recognition for MUC-6. The

article describes how the Alembic computer program worked to achieve the goals of the conference.

36. As set forth in the attached claim chart, this article clearly anticipates claim 27. It discloses plural subject matter specific tags, and the tags are used in pairs with beginning and end tags. The article contains extensive discussion of rules that allowed the system to discriminate between different data types. The system would receive a record, and the Alembic computer program would examine the record according to the search rules to identify segments of the record that include information of each type. When such segments are located, the program would insert the tags for that information type around the segment. An example of the resultant tagged text is found on page 145 of the article:

> *Yesterday, <none>McCann</none>made official what had been widely anticipated: <ttl>Mr.</ttl><person>James</person>, <num>57</num>years old, is stepping down as <post>chief executive</post> on <date>July 1</date> and will retire as <post>chairman</post> at the end of the year.*

37. Krupka (Krupka, G., SRA: Description of the SRA System as Used for MUC-6, Proc. of the 6th Conf. on Message Understanding, pp. 221-35, Columbia, Maryland (Nov. 6-8 1995)): This article describes another way in which Named Entity Recognition was achieved for MUC-6. The article describes how a commercial software product called NameTag was used in combination with another program called HASTEN. NameTag recognizes proper names and other key phrases in text, which it tags with SGML tags. The results of NameTag were fed into HASTEN, which extracted and generated management succession templates. See Figure 1.

38. As set forth in the attached claim chart, this article clearly anticipates claim 27. The NameTag program is described in detail beginning at page 232. This program uses “recognition rules” to recognize different types of information, such as personal names, organization names, and places. The output is an annotated SGML file, with the recognized information tagged.

While the article does not provide an explicit example of the format of the tags, given that the objective of MUC-6 was to use subject matter specific tag pairs with beginning and end tags, and given that the introduction of the article states that the specification was achieved (“SRA adapted a subset of NameTag’s capabilities to the MUC-6 specification,” at 221), the article would clearly have been understood at the time by a person of skill in the art to disclose subject matter specific tag pairs with beginning and end tags. In any event, it clearly would have been obvious to use subject matter specific tag pairs with beginning and end tags, since that was commonplace at the time, as evidenced by the MUC-6 Named Entity Task Definition.

39. Mills: As set forth in the attached claim chart, this article clearly anticipates claim 27. It discloses plural subject matter specific tags, such as NAME, DATE, and LAND and the tags are used in pairs with beginning and end tags. A record would be received, and reviewed. Common-sense rules were then used to identify segments of the record that include information of each type. When such segments were located, the record was annotated by inserting the tags for that information type around the segment. An example of the resultant tagged text is found on page 1319 of the article:

```
<DOCUMENT>
<IDENTIFIER> 0140.00552 </IDENTIFIER>
<TYPE> PRO STAC8/289/30 in ERO transcript </TYPE>
<DATE date=25.11.1606> 25.11.1606 </DATE>
<TEXT>
<NAME identity=H206 name=“Richard Harlakenden”> Rich Harlakenden </NAME>
forced to enter a bill for forcible detainment of <NAME identity=F314
name=“Wm Ford”> Wm Forde </NAME> and others <NAME identity=F314 name=“Wm
Ford”> Wm Forde </NAME> found guilty of forcible detainment of <LAND
identity=152 name=“Chalkney Mill”> Chalkney Mill </LAND> and <NAME
identity=H206 name=“Richard Harlakenden” > Harlakenden </NAME> was put in
possession by the sheriff through a writ of restitution
</TEXT>
</DOCUMENT>
```

40. I understand from Dr. Nirenburg’s report (at 82-84) that his sole counter-argument to my position that Mills anticipates claim 27 is his view that claim 27 describes an “automatic”

process (both "automatically" examining a record, and "automatically" inserting tags), whereas the tags in the above-quoted sample were inserted manually. While this is a claim construction issue that I understand will be addressed by the court, I do not see anything in claim 27 that restricts the claim to "automatic" processes, or that excludes "manual" processes. Moreover, in the event that claim 27 is so-construed, it would have been obvious to use automatic processing of the underlying data records in Mills to add subject matter specific tags. As the above discussion shows, it was well-known to have computer programs that would automatically examine data records to identify precisely the types of information recognized in Mills, and add subject matter specific tag pairs before and after the recognized segments of text. One clearly would be motivated to make this change, since it would be far easier and faster to add tags automatically rather than manually.

4. '321 Patent, Claim 86:

41. Claim 86 reads as follows:

> ***Claim 86.*** *A method for use with an application wherein specifying references (SRs) in one record to other records which are selectable to access the other records are visually distinguished from other record information so as to indicate selectability, the method also for use with a system which enables a user to designate and also select SRs where designation comprises pointing to an SR without selection and, wherein a seemingly general SR is modified by other record information which renders the SR relatively specific, the method for indicating the specific nature of an SR prior to selection and comprising the steps of:*
>
> *when an SR is designated, indicating the specific nature of the SR*

42. As an initial matter, this claim is indefinite in at least two respects other than indicated in my original expert report. The claim, and the '321 patent, do not explain in any meaningful sense what it means for an SR to be "seemingly general." Nor do they explain how other record information renders the "seemingly general" SR "relatively specific." In my view, the claim does not set forth the boundaries of the claim scope to a person of skill in the art, and thus in indefinite.

43. For a comparison of the disclosure of each prior art reference to the asserted claims on a limitation by limitation basis, please refer to the claim charts attached as Appendix C to my original report, and Appendix C of this supplemental report.

44. Gennaro (United States Patent No. 5,742,768): This patent discloses a method and system in which a user can hover a cursor over an item of interest to obtain more information about the item. As shown in Figs. 2A and 2B, if a user moves the cursor over a hot spot 44, a menu 46 pops up. The user can then move the cursor over the menu entries, in which case a URL 38 for each menu item is displayed in the lower left hand corner of the browser.

45. As set forth in the attached claim chart, this patent clearly anticipates claim 86. There are actually two different SRs disclosed in this reference. Each of the hot spots 44 is an SR. Each of the items in menu 46 is also an SR.

46. As to the hot spots, these are visually distinguished from other record information because, e.g., they are adjacent to a circle, that converts to a different graphic when the cursor is positioned over the hot spot. The user can point to the hot spot without selection, by "mousing over" (using the mouse to move the cursor over the hot spot), in which case the menu pops up. The menu items further detail the specific topics of information available for each hot spot, and thus constitute other record information that renders the SR (the hot spot text) more specific. Thus, when the user mouses over the hot spot, thereby designating the SR, the specific nature of the hot spot is indicated by displaying the menu. Not only is this an implementation as described in the claim, but it is also the same as the embodiment of claim 86 described in the patent at columns 14 and 15 and as shown in, for example, Fig. 26 of the patent.

47. As to the menu items, these are visually distinguished from other record information because they appear in the menu. The user can point to a menu item without selection, by mousing over, in which case the URL for each menu item (item 38) is displayed in the lower left hand corner of the browser. The URL constitutes other record information that renders the SR (the menu item text) more specific. Thus, when the user mouses over the menu

item, thereby designating the SR, the specific nature of the menu item is indicated by displaying the URL.

48. Netscape/Internet Explorer: The functionality disclosed in Gennaro wherein the URL of a hyperlink is displayed in the lower left hand corner of the browser was also clearly disclosed in both the prior art Netscape Navigator browser and the prior art Internet Explorer browser. A hyperlink essentially has two components, an anchor and a URL. The anchor is the text that is displayed in the browser window. The URL is the address of the webpage that is retrieved if a user clicks on the anchor. The anchor is displayed in a manner such as to indicate that it is a hyperlink, such as in boldfaced, underlined blue text.

49. This functionality clearly anticipates claim 86, for the reasons set forth in my claim charts. The SR is the anchor text. The anchor text is visually distinguished from other text to indicate selectability. If the anchor text is selected, other records are accessed. A user can designate an SR (the anchor text) without selecting it. The anchor text is "seemingly general," in that the same anchor text can refer to any one of a number of target webpages. The underlying URL information, however, modifies the SR/anchor to render it relatively specific, i.e., it indicates the URL of one and only one webpage that will be called up if the user selects the anchor. The specific nature of the SR/anchor is displayed in the lower left hand corner of the browser if the user designates the anchor by mousing over it.

50. Myka: This reference likewise discloses the claim 86 feature of providing additional information about a link by mousing over it. Myka uses a pattern-matching technique to identify candidate links, and then further uses information about the context in which the candidate links appear in the text (such as layout information) to further define the link. Once the link has been defined, if a user moves the cursor over the link, which is visually distinguished from the surrounding text by being boxed, other record information about the link, and that modifies the link and renders it relatively specific, is displayed in a window on the right hand side of the window. For example, this information can be about the target page accessed by clicking the link, or can be actions that can be performed with the link. Thus, as in claim 86, the

link can either be selected to retrieve the target document, or can be designated to indicate the specific nature of the link, either by providing information about the target, or by listing actions for the link.

51. To the extent not anticipated, it is my opinion that the limitations required by this claim would have been obvious to a person of ordinary skill in the art at the time of the patent. For an analysis of obviousness combinations on a limitation by limitation basis, please refer to the attached claim charts.

B. Further Considerations Regarding Obviousness

52. To the extent that any reference that I have described in Appendix C to my original report or this supplemental report as anticipatory is found to not disclose a particular limitation, it is my opinion that all of the asserted claims would still have been obvious to a person of ordinary skill in the art at the time of the patent in light of these references. Given the close overlap in subject matter, purpose, goals, and objectives, any of these references could have been readily combined with any other reference to supply any missing limitation(s).

53. Many are also directed to the same problem of automatic creation of an information retrieval hypertext. This is demonstrated by the fact that many of the references build on or reference work disclosed in other references, or are designed to work with or as add-ons to the same programs or types of programs, referencing for example word processors such as Microsoft Word or Corel WordPerfect.

54. As for secondary considerations, I remain unaware of any that apply for any asserted claim of any of HyperPhrase's patents, where the basis for the secondary considerations pertains to the claimed subject matter. There was no long-felt need for the features of these claims, as is demonstrated by the existence of systems disclosing all of the limitations of each claim, as claimed, in many cases years before any of HyperPhrase's asserted patents was filed. Moreover, these features would have been straightforward to implement, without undue experimentation, by a person of ordinary skill in the art at the time of the patent, as demonstrated

by the existence of prior art references and systems disclosing all of these features in many cases years prior to the filing of the asserted patents' applications. Similarly there was no failure by others, as demonstrated by the references and systems described in detail in the Exhibits attached in Appendix C.

55. In fact, I recall attending DARPA workshops and other meetings prior to and in 1995 in which demonstrations were given where the systems were dynamically processing documents and inserting links. Although these demonstrations involved real-time recognition, highlighting, and hypertext linking of keyword phrases such as company names or product names, the purpose of the demonstration was not to show that recognition of keywords and creation of links was possible. That was already common knowledge. Rather, these demonstrations were for the purpose of showing how quickly more complicated algorithms for name recognition could process the text, in order to give those at the conference an understanding of how quickly a given database, which could be quite large, could be processed.

56. Nor am I aware of any commercial embodiments of the claimed invention. To the extent that HyperPhrase intends to rely on Google's AutoLink technology as allegedly supporting an argument of commercial success, it has not shown that any alleged commercial success was in any way due to the features of AutoLink allegedly covered by the asserted claims, as opposed for example to other aspects of AutoLink or to the myriad other functions available on the toolbar of which it is a part.

57. This report contains a summary of the opinions I have reached to date. I reserve the right to render additional opinions, to supplement or amend the opinions in this report and to provide additional opinions based upon ongoing analysis, or as may be required by events that may occur in the course of this litigation. In connection with my anticipated trial testimony in this action, I may use as exhibits various documents produced, or testimony given, in this litigation which refer or relate to the matters set forth in this report. In addition, before my trial testimony, I may create certain demonstrative exhibits to be used at trial. Finally, I reserve the

right to demonstrate for the Court and jury the operation of any and all of the prior art references or systems discussed in this report.

Dated: February 11, 2008

By: ____________________________
W. Bruce Croft, PhD.

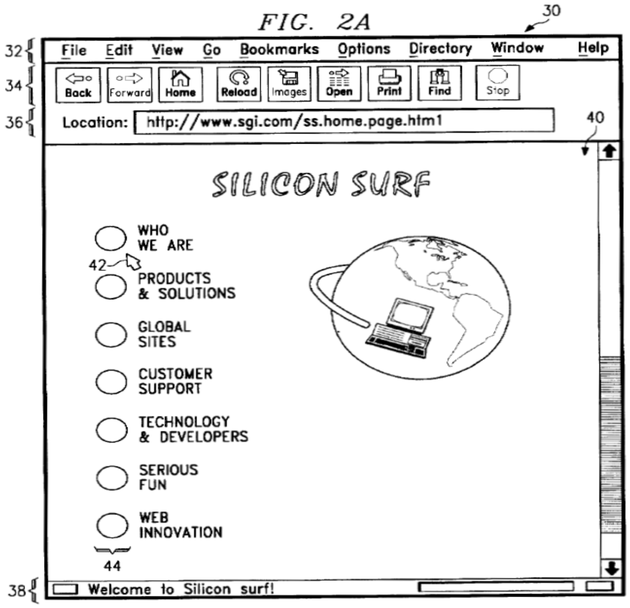
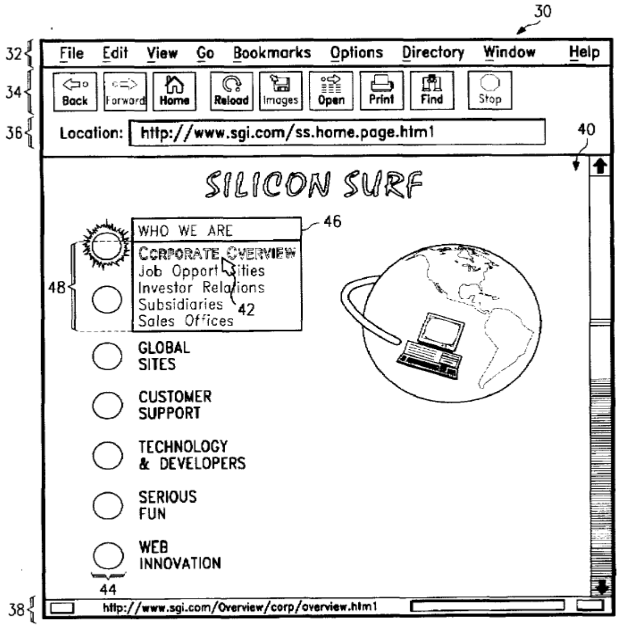
EXHIBIT 2a

APPENDIX C-1

SUPPLEMENTAL REPORT INVALIDITY ANALYSIS

U.S. Patent No. 5,742,768 to Gennaro et al., entitled "System and Method For Providing and Displaying a Web Page Having an Embedded Menu," (filed Jul. 16, 1996; issued Apr. 21, 1998) [hereinafter, "Gennaro"]

U.S. PATENT NO. 6,516,321	DISCLOSURE
Claim 86. [i] A method for use with an application, wherein specifying references (SRs) in one record to other records which are selectable to access the other records are visually distinguished from other record information so as to indicate selectability,	Gennaro discloses a method to be used with an application, specifically a web browser, where SRs that are capable of accessing other records are made selectable and visually distinguishable from other record information. Hot spots 44 are one example of SRS. The individual items in menu 46 can also be considered SRs. For example: "A method for providing a web page (26) having an embedded menu (46) to a web browser (24) and for displaying the web page (40) to a user of the web browser (24) are provided. [Abstract; see, e.g., 2:4-7] "FIG. 2B shows an embedded menu 46 in displayed web page 40 which has been invoked by positioning of pointer 42 over the upper hot spot 44. In the illustrated example, selection of the upper hot spot 44 is indicated by highlighting that hot spot 44 with a halo as shown. Embedded menu includes a banner that matches the text ('WHO WE ARE') that was associated with the selected hot spot 44 in FIG. 2A. Embedded menu 46 also includes a number of links 48, each providing a link to another web page or resource. The links 48 provided by embedded menu 46 may or many not by URLs directly accessible without initially passing through the initial displayed web page." [4:30-41 ;FIG. 2A; FIG. 2B] "When pointer 42 is moved outside of embedded menu 46, embedded menu 46 will be removed and displayed web page 40 will again look as shown in FIG. 2A. The user can move pointer 42 over any hot spots 44 and invoke an associated embedded menu, each of which would provide multiple links to other web pages or resources." [4:54-59; FIG. 2A] "In one implementation, the web page has one or more hot spots. When a pointer is positioned over one of these hot spots, a corresponding menu is displayed to provide links to multiple additional web pages. A user can then select a link by positioning the pointer over one of the links and initiating an action such as by clicking a mouse button." [2:47-53; FIG. 2B]

	 FIG. 2A 30 32 34 36 38 40 42 44 File Edit View Go Bookmarks Options Directory Window Help Back Forward Home Reload Images Open Print Find Stop Location: http://www.sgi.com/ss.home.page.htm1 SILICON SURF WHO WE ARE PRODUCTS & SOLUTIONS GLOBAL SITES CUSTOMER SUPPORT TECHNOLOGY & DEVELOPERS SERIOUS FUN WEB INNOVATION Welcome to Silicon surf! [FIG. 2A]  30 32 34 36 38 40 42 44 46 48 File Edit View Go Bookmarks Options Directory Window Help Back Forward Home Reload Images Open Print Find Stop Location: http://www.sgi.com/ss.home.page.htm1 SILICON SURF WHO WE ARE CORPORATE OVERVIEW Job Opportunities Investor Relations Subsidiaries Sales Offices GLOBAL SITES CUSTOMER SUPPORT TECHNOLOGY & DEVELOPERS SERIOUS FUN WEB INNOVATION http://www.sgi.com/Overview/corp/overview.htm1 [FIG. 2B] "If the mouse pointer remains inside the embedded menu, then, in step 80, the applet highlights the selected link within the embedded menu and displays the destination URL associated with the selected link." [6:7-10; FIG. 4]
[ii] the method also for use with a system which enables a user to designate and also select SRs where designation comprises pointing to an SR without selection and,	Gennaro disclose a method implemented by a system which allows a user to designate SRs by mousing over the SRs to determine more specific information about them, without selection. See discussion above in connection with feature [i]. See also: "A method for providing a web page (26) having an embedded menu (46) to a web browser (24) and for displaying the web page (40) to a user of the web browser (24) are provided. [Abstract]

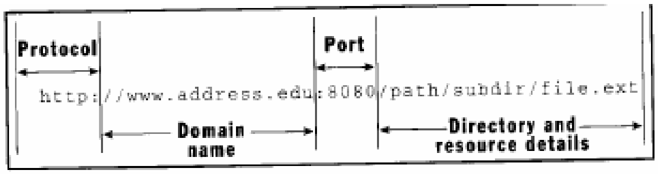
	"FIG. 1 is a block diagram of a web server 12 and a web browser 14 in an IP (internet protocol) network 16. IP network 16 can be, for example, the public Internet or a private intranet and host system 12 and client system 14 can communicate across IP network 16 using a hyper-text transfer protocol (HTTP)." [3:20-25] "According to a further aspect of the invention, a host system for providing a client system for displaying a web page having an embedded menu are provided. The host system includes a data storage device, a memory and a processor and executes a web server for packaging and transmitting the web page and applet." [2:27-32] "A user can reposition a pointer over each hot spot to invoke each embedded menu and be provided with multiple links at one or more levels within the web site."[2:58-60; 4:64-66] "A user can reposition a pointer over each hot spot to invoke each embedded menu and be provided with multiple links at one or more levels within the web site." [2:58-60]
[iii] wherein a seemingly general SR is modified by other record information which renders the SR relatively specific,	Gennaro discloses a method implemented by a system to modify a general SR which renders the SR relatively specific, as discussed above in connection with features [i] and [ii]. For example, when the "hot spots" are the SRs the embedded menu options, such as "Corporate Overview" render that SR relatively specific. When the embedded menu entries, such as "Corporate Overview" are the SRs, the URL, item 38 renders that SR relatively specific, as is displayed in FIG. 2B in the lower left hand corner of the browser. *See also*: "In FIG 2B the 'Corporate Overview' link is selected by the positioning of pointer 42 over that portion of embedded menu 46, and status bar 40 reflects the URL associated with 'Corporate Overview' link. If desired, the user of web browser 23 can link 'Corporate Overview' information by initiating an action, for example by clicking a mouse button, while pointer 42 is in this position. The user could also move pointer 42 elsewhere in embedded menu 46 to select and initiate one of the other links. In other embodiments, embedded menu 46 can have multiple levels of menus accessible through initial menu options." [4:42-53]
[iv] the method for indicating the specific nature of an SR prior to selection and comprising the steps of:	Gennaro discloses a method implemented by system to indicate the specific nature of an SR prior to selection, as discussed in connection with the above feature [ii].
(a) when an SR is designated, indicating the specific nature of the SR.	Gennaro discloses a method implemented by system to indicate the specific nature of an SR once designated, as discussed in connection with the above feature [ii].

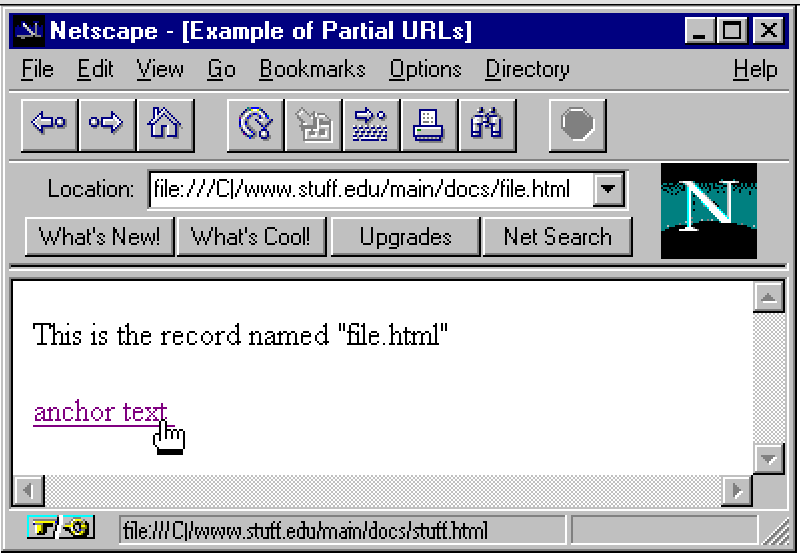
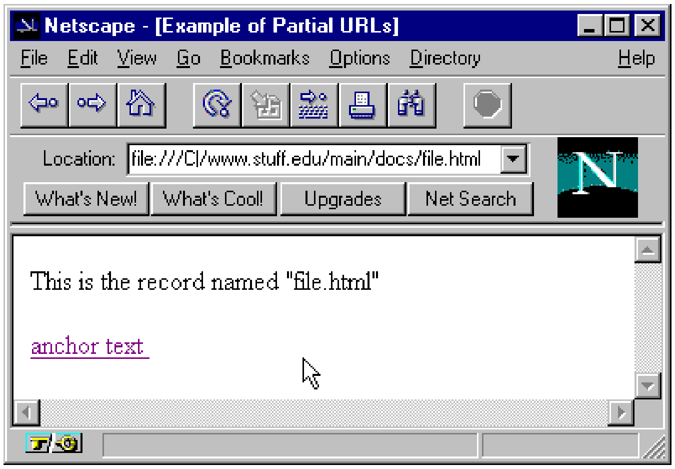
EXHIBIT 2b

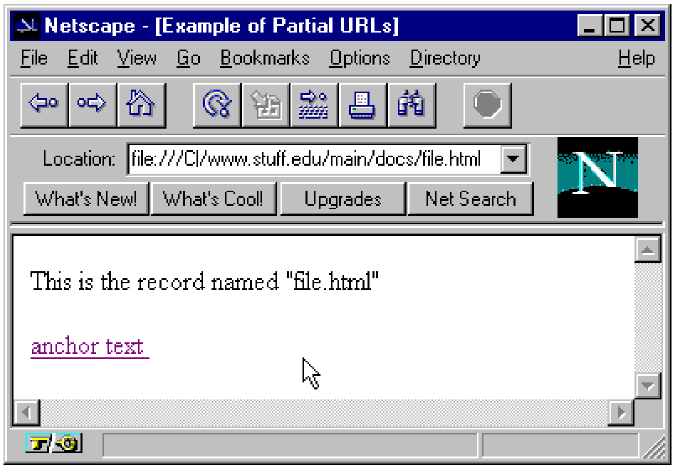
APPENDIX C-2

SUPPLEMENTAL REPORT INVALIDITY ANALYSIS

Graham, Ian S., *HTML Sourcebook: A Complete Guide to HTML*, pp. 167-68 (John Wiley & Sons. Eds., Mar. 1995), ISBN: 0-471-11849-4 [hereinafter "The HTML Sourcebook"]

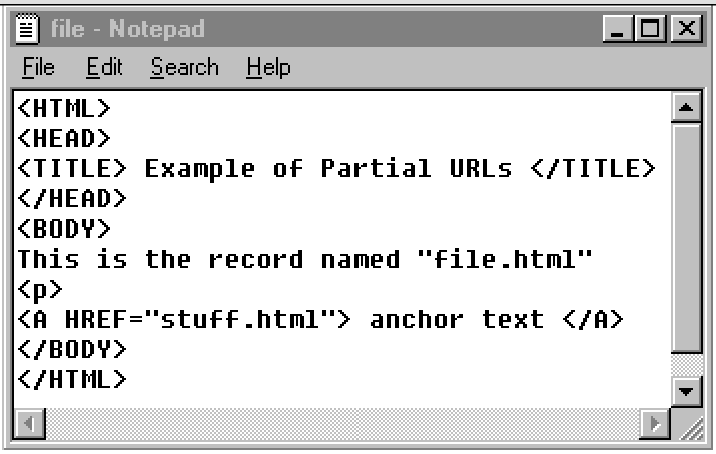
U.S. PATENT NO. 5,903,889	DISCLOSURE
Claim 1. A computer system with a plurality of data records on a plurality of databases, and a standardized format for addressing said data records, said computer system comprising:	A web browser, which is designed to be used in conjunction with the World Wide Web, can be used to retrieve data records on a plurality of databases, e.g., on servers spread across the Internet, where there is a standardized format for addressing the records, all of which was obvious at the time. *See, e.g.,* Thistlewaite (PasTime) at 171; *see also* Hall (Rethinking Hypermedia) at pp. 38, 47. "What resources can URLs point to? They can be . . . . on your computer or anywhere on the Internet. [HTML Sourcebook at xii.]  Protocol Port http://www.address.edu:8080/path/subdir/file.ext Domain name Directory and resource details **Figure 3.1** Example of a URL. [HTML Sourcebook at 166; *see also id.* at p. 42.]
(a) a user interface having an interactive display program for requesting one of said data records and displaying a plurality of interface supported data formats;	The corresponding structure for this element is a web browser. *See* '889 patent, col. 2, ll. 50-57. The HTML Sourcebook discloses viewing files using a web browser. A browser includes a user-interface (e.g., the interface shown below) for requesting records (e.g. **file.html**) that can display data formats such as text, hyperlinks, graphics, multimedia and other formats. "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape." [HTML Sourcebook at xii.] "HTML contains commands, called tags, to mark text as headings, paragraphs, lists . . . . It also has tags for including images within the documents, for including fill-in forms that accept users input, and, most importantly, for including hypertext links connecting the document being read to other documents or Internet resources . . . ." [HTML Sourcebook at xi.]

U.S. PATENT NO. 5,903,889	DISCLOSURE
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.]
(b) means for receiving a reference to a first data record from said interactive display program;	The corresponding structure for this element is a web browser, and in particular the field that allows a user to type in a URL to request an HTML file (a first data record). *See* '889 patent, col. 2, ll. 50-57. The HTML Sourcebook discloses that a user can enter a URL in the "Location" field of a browser (shown below) to reference and retrieve a data record. The HTML Sourcebook further discloses receiving a document through a web browser. "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML hypertext anchors and the URLs they contain . . . ." [HTML Sourcebook at xii.] "Uniform Resource Locators, or URLs, which are the scheme by which Internet resources are addressed in the WWW ." [HTML Sourcebook at ix.]  Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text [Netscape Navigator Ver. 1.22 – Viewing **file.html**.]
(c) means for retrieving said first data record;	The corresponding structure for this element is a web browser, and in particular the capability to make requests for HTML files in accordance with a URL. *See* '889 patent, col. 3, ll. 44-54

U.S. PATENT NO. 5,903,889	DISCLOSURE
	The HTML Sourcebook discloses a web browser for retrieving records. "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape." [HTML Sourcebook at xii.] "Suppose you originally access the document **file.html** using the full URL: `http://www.stuff.edu/main/docs/file.html"` [HTML Sourcebook at 167.]  Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text [Netscape Navigator Ver. 1.22 – Viewing **file.html**.]

(d) means for parsing said first data record to identify a reference to a second data record;	The corresponding structure for this element is a generic, black-box program that parses incoming data records. *See* '889 patent, col. 8, ll. 37-40. The details of the parser are not adequately disclosed in the specification.
	The HTML Sourcebook discloses the web browser parsing the document to identify references (e.g., URLs) contained therein:
	"You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . ***Browsers understand HTML hypertext anchors and the URLs they contain*** . . . ." [HTML Sourcebook at xii (emphasis added).]
	"Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>`
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html." [HTML Sourcebook at 167-68.]

U.S. PATENT NO. 5,903,889	DISCLOSURE

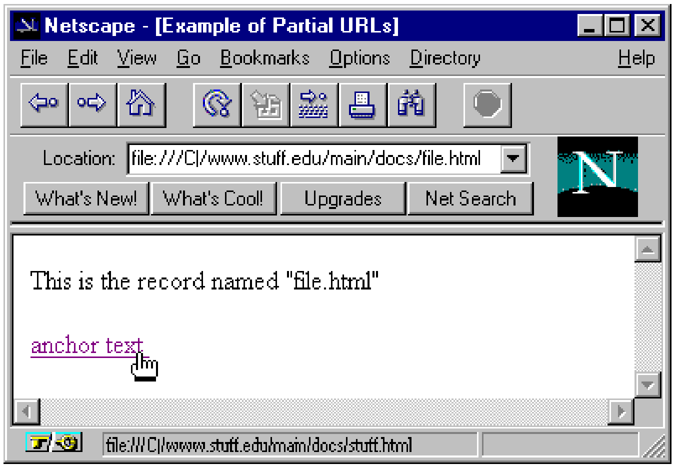
	
	
	`<HTML>`
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**File.html**.]

(e) means for modifying said reference to said second data record to create an address, said address being operable to retrieve said second data record; and	The corresponding structure for this element is a generic, black-box program that modifies the reference in the first data record. *See* '889 patent, col. 9, ll. 1-11. The details of this program are not adequately disclosed in the specification.
	
	The HTML Sourcebook discloses a web browser which renders HTML documents for display to the user:
	
	"You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML hypertext anchors and the URLs they contain . . . ." [HTML Sourcebook at xii (emphasis added).]
	
	The HTML Sourcebook discloses resolving a partial URL to a complete URL compliant with the URL addressing format, where that resolved URL is operable to retrieve the referenced data record:
	
	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>`
	
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access **file.html**. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`
	
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html." [HTML Sourcebook at 167-68.]
	
	"Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`"
	[HTML Sourcebook at 167-68.]
	
	To the extent that this element is construed to require modification of the underlying first data record, it would have been obvious to do so. *See, e.g.,*

<table>
<tr><th>U.S. PATENT NO. 5,903,889</th><th>DISCLOSURE</th></tr>
<tr><td></td><td>

PasTime at 166.

When the user clicks on the resolved URL (that the URL has been resolved is shown in the lower left portion of the web browser window below), the target record is retrieved:



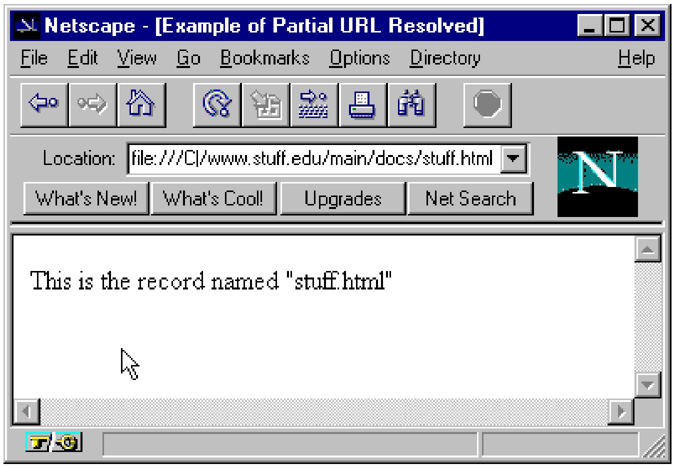
Netscape - [Example of Partial URLs]
File Edit View Go Bookmarks Options Directory Help
Location:
What's New! What's Cool! Upgrades Net Search
N
anchor text

Location: file:///C|/www.stuff.edu/main/docs/file.html

This is the record named "file.html"

file:///C|/www.stuff.edu/main/docs/stuff.html

[Navigator – Hovering cursor over hyperlink in **file.html**.]



Netscape - [Example of Partial URL Resolved]
File Edit View Go Bookmarks Options Directory Help
Location:
What's New! What's Cool! Upgrades Net Search
N

Location: file:///C|/www.stuff.edu/main/docs/stuff.html

This is the record named "stuff.html"

[Navigator – After user has traversed hyperlink to file **stuff.html**.]

</td></tr>
<tr><td>

(f) means for sending said modified first data record to said interactive display program.

</td><td>

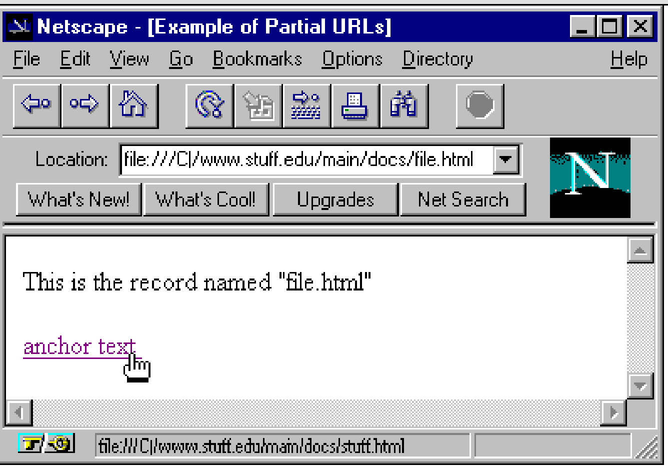
The corresponding structure for this element is a generic, black-box program that sends the modified first data record to the browser. *See* '889 patent, col. 9, ll. 12-15. The details of this program are not adequately disclosed in the specification.

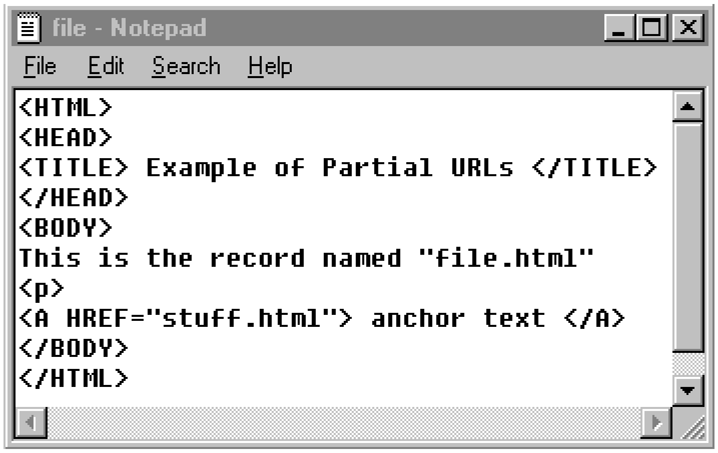
HTML Sourcebook discloses a web browser for displaying a rendered HTML page to the user.

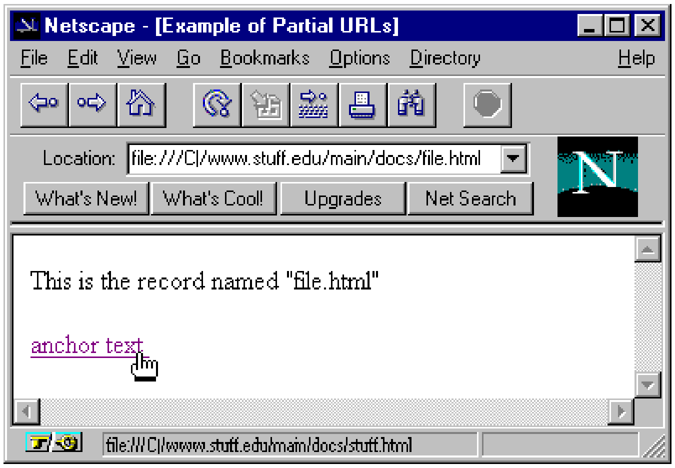
"You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape." [HTML Sourcebook at xii.]

As can be seen below, the rendered page includes the modified "anchor text":

</td></tr>
</table>

U.S. PATENT NO. 5,903,889	DISCLOSURE
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help Location: What's New! What's Cool! Upgrades Net Search file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.] "The element marking a hypertext link is called an A, or *anchor*, element, and the marked text is referred to as a *hypertext anchor*. The area between the beginning <A> and ending </a> tags becomes a *hot* part of the text. With Mosaic or Mac Web, this section of text is displayed with an underline and usually in a different color . . . . Placing the mouse over this region and clicking the mouse button . . . causes the client to access the indicated document or other internet resource." [HTML Sourcebook at 21.]

Claim 7. The computer system of claim 1, wherein said reference to said second data record comprises a keyword phrase.	The HTML Sourcebook discloses using a reference to a second data record, where that reference comprises the keyword phrase "anchor text.":
	
	"You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape." [HTML Sourcebook at xii.]
	
	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>"`
	[HTML Sourcebook at 167.]
	
	
	
	`<HTML>`
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**File.html**.]

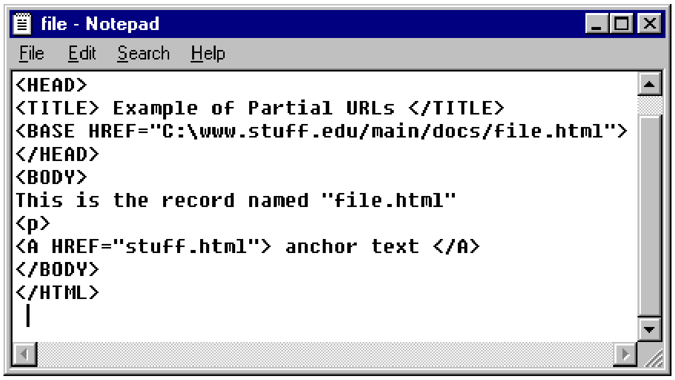
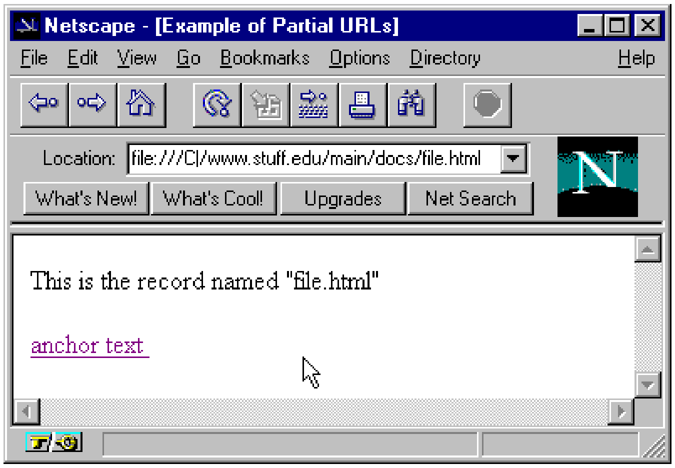
U.S. Patent No. 5,903,889	Disclosure
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search N Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.]

U.S. Patent No. 6,516,321	**Disclosure**

Claim 1. A method for identifying a referenced record referenced in a referencing record wherein the referenced record is referenced in the referencing record by at least a combination including a data reference (DR) and a modifier reference (MR), the method comprising the steps of:	The HTML Sourcebook discloses a method for identifying a referenced record from within a referencing record, through the combination of a data reference (DR) (e.g., a partial URL) and a modifier reference (MR) (e.g., a BASE URL).
	
	"You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML *hypertext anchors* and the URLs they contain . . . ." [HTML Sourcebook at xii.]
	
	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>`
	
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`
	
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html. Other equivalent partial URLs are then:
	/main/docs/stuff .html
	//www.stuff.edu/main/docs.html
	
	The former appropriates http:/www.stuff.edu from the current URL to complete the reference, while the latter appropriates only the http: part from the base URL of the current document.
	
	You can also use partial URLs to reference files in other directories; for example, from the example **file.html** the relative URL:
	../../main.html
	
	indicates the file **main.html** in the root HTTP directory, namely:
	http://www.stuff .edu/main.html
	
	Partial URLs are very useful when constructing large collections of documents that will be kept together. Of course relative URLs become invalid if a document is moved to a new directory or a new Internet site. This problem can be mitigated using the BASE element of the HTML, which is used to record the correct BASE URL of a document. If the document is moved, all relative URLs are determined relative to the URL recorded by the BASE element." [HTML Sourcebook at 167-68; *see also id.* At 86-88 (BASE URL).]

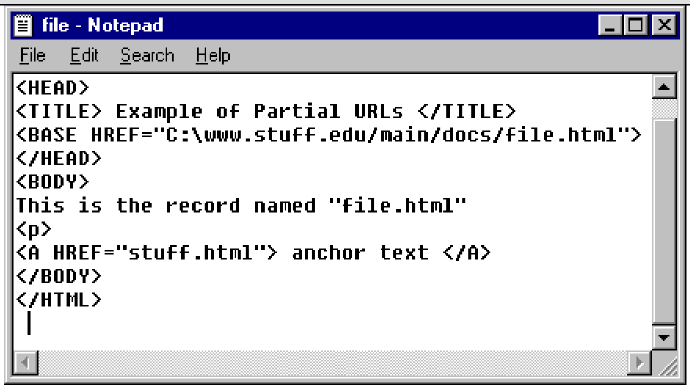
(i) receiving the referencing record;	The HTML Sourcebook discloses receiving a document through a web browser: "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML *hypertext anchors* and the URLs they contain . . . ." [HTML Sourcebook at xii.]

U.S. PATENT NO. 6,516,321	DISCLOSURE

	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>"`
	[HTML Sourcebook at 167.]
	
	
	
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`<BASE HREF="C:\www.stuff.edu/main/docs/file.html">`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**File.html**.]
	
	
	 Netscape - [Example of Partial URLs]
	File Edit View Go Bookmarks Options Directory Help
	What's New! What's Cool! Upgrades Net Search
	N
	Location: file:///C|/www.stuff.edu/main/docs/file.html
	This is the record named "file.html"
	anchor text
	[Navigator – Viewing **file.html**.]
	
	"Uniform Resource Locators, or URLs, which are the scheme by which Internet resources are addressed in the WWW ." [HTML Sourcebook at ix.]

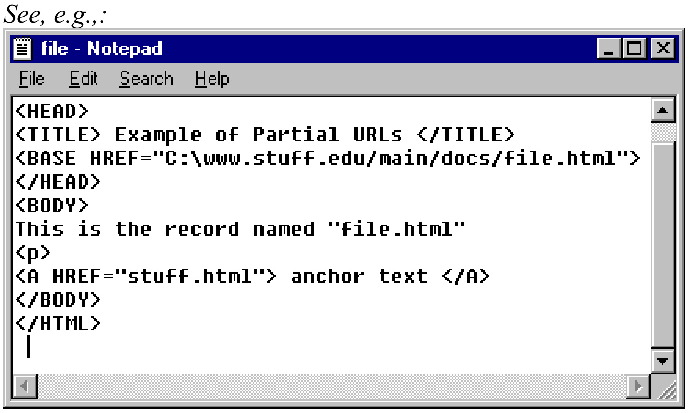
(ii) analyzing the referencing record to identify a DR, when a DR is identified:	The HTML Sourcebook discloses a web browser analyzing a document (e.g., an HTML document) to find a partial URL. "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML *hypertext anchors* and the URLs they contain . . . ." [HTML Sourcebook at xii.] Upon receiving a record, the browser would analyze the record for any HTML references (e.g., partial URLs) to other records.

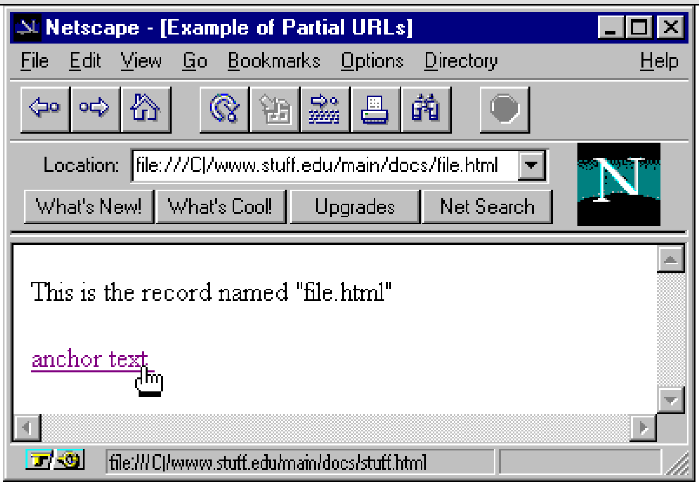
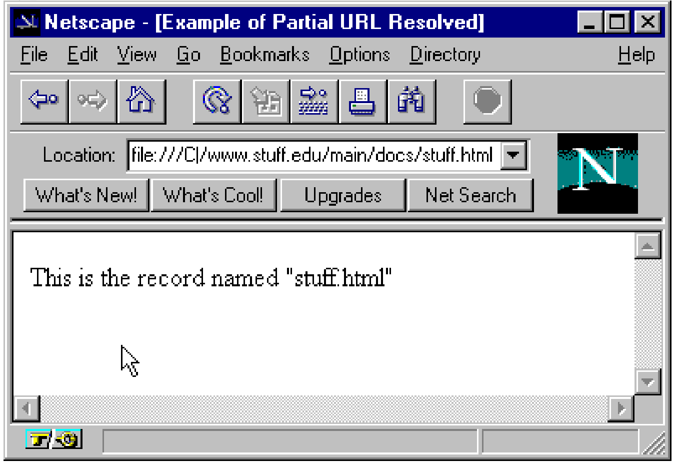
U.S. PATENT NO. 6,516,321	DISCLOSURE

	
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`<BASE HREF="C:\www.stuff.edu/main/docs/file.html">`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**file.html**.]

(a) identifying an MR rule set (MRRS) specifying the relationship between an MR and the DR;	If it finds a partial URL, the HTML Sourcebook discloses rules used to resolve partial URLs, such as looking to see if a BASE URL is defined within the record, and if it is defined, using the BASE URL to resolve the partial URL.
	
	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>`
	
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`
	
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html. Other equivalent partial URLs are then:
	/main/docs/stuff .html
	//www.stuff.edu/main/docs.html
	
	The former appropriates http:/www.stuff.edu from the current URL to complete the reference, while the latter appropriates only the http: part from the base URL of the current document.
	
	You can also use partial URLs to reference files in other directories; for example, from the example **file.html** the relative URL:
	../../main.html
	
	indicates the file **main.html** in the root HTTP directory, namely:
	http://www.stuff .edu/main.html
	
	Partial URLs are very useful when constructing large collections of documents that will be kept together. Of course relative URLs become invalid if a document is moved to a new directory or a new Internet site. This problem can be mitigated using the BASE element of the HTML, which is used to record the correct BASE URL of a document. If the document is moved, all relative URLs are determined relative to the URL recorded by the BASE element." [HTML Sourcebook at 167-68; *see also id.* At 86-88 (BASE URL).]

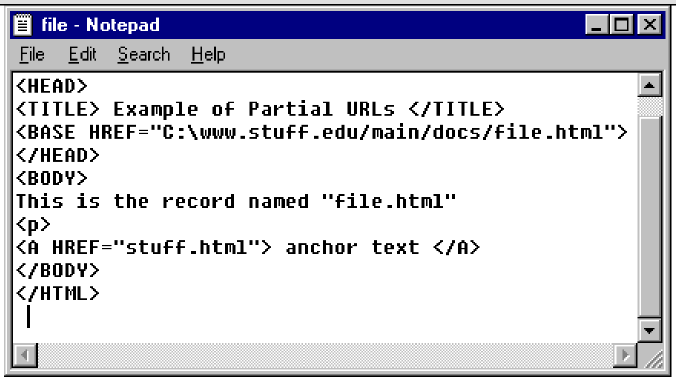
(b) analyzing the referencing	The HTML Sourcebook discloses a web browser that analyzes the record in

U.S. Patent No. 6,516,321	Disclosure
record in accordance with the MRRS to identify the existence of the MR and, when the MR is identified;	accordance with the set of rules to find a BASE URL. *See id.* If a BASE URL is found;
(c) identifying the referenced record associated with the DR/MR combination.	The web browser identifies the referenced record associated with the DR/MR combination (e.g., uses the partial URL/BASE URL combination to identify a record specified by the fully resolved URL). "Thus, the partial URL stuff.html is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then: http://www.stuff.edu/main/docs/stuff.html which indicates, as expected, that stuff.html is on the same server and in the same directory as file.html. … The former appropriates http:/www.stuff.edu from the current URL to complete the reference, while the latter appropriates only the http: part from the base URL of the current document. You can also use partial URLs to reference files in other directories; for example, from the example **file.html** the relative URL: ../../main.html indicates the file **main.html** in the root HTTP directory, namely: http://www.stuff .edu/main.html Partial URLs are very useful when constructing large collections of documents that will be kept together. Of course relative URLs become invalid if a document is moved to a new directory or a new Internet site. This problem can be mitigated using the BASE element of the HTML, which is used to record the correct BASE URL of a document. If the document is moved, all relative URLs are determined relative to the URL recorded by the BASE element." [HTML Sourcebook at 167-68.] *See, e.g.,:*  <HEAD> <TITLE> Example of Partial URLs </TITLE> <BASE HREF="C:\www.stuff.edu/main/docs/file.html"> </HEAD> <BODY> This is the record named "file.html" <p> <A HREF="stuff.html"> anchor text </A> </BODY> </HTML> [**File.html**.]
Claim 24. The method of claim 1 further including the step of linking the record reference to the referenced record.	When a resolved partial URL is displayed, the referenced record is hypertext linked to referencing record. This is shown, for example, in the file.html file when viewed in a web browser. The underlined "anchor text" is a hypertext link to the referenced record.

U.S. PATENT NO. 6,516,321	DISCLOSURE
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.] Below is the referenced record, which is retrieved after clicking on the hypertext link in **file.html**:  Netscape - [Example of Partial URL Resolved] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/stuff.html This is the record named "stuff.html" [**Stuff.html**.]
Claim 27. A method to be used with a rule set including subject matter specific tag pairs and corresponding search rules, a separate tag pair for each of a plurality of different information types and a separate search rule for each pair, each pair including a begin tag and an end tag, the method comprising the steps of:	The HTML Sourcebook provides an extensive description of rules used to parse an HTML document using different rule sets to insert subject matter specific tag pairs for different types of information (e.g., header tags, anchor tags, body tags, etc…). A detailed description of subject matter specific tagging is provided in the HTML Sourcebook at pages 76 through 159, incorporated herein by reference. "The difference between an HTML document and a simple text document lies in the HTML markup *tags*. These are the portions of text surrounded by the less than and greater than signs (< . . . >) and are the *instructions* that tell the browser what each part of the document means. For example, the tag <H1> indicates the *start* of a heading of level 1, while the </H1> tag marks the *end* of a heading of level 1. Thus the text string <h1> This is the Heading </H1> Marks the string "This is the Heading" as a level one heading (there are six possible headings levels, from H1 to H6). Note how the forward slash inside the tag indicates an end tag…. An HTML document is described as being composed of elements. The string <h1> This is the Heading </H1>

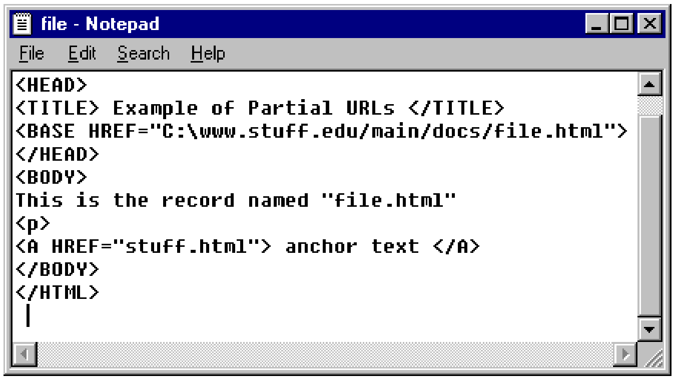
U.S. PATENT NO. 6,516,321	DISCLOSURE
	Is then an H1 element, consisting of an H1 start tag, the enclosed text, and an H1 end tag. You will often see an H1 element referred to as the *container* of a heading." [HTML Sourcebook at 2 (emphasis in original).] "Sometimes, elements take *attributes*, which are much like variables and are usually assigned values that define special characteristics of the element. An example is the IMG element, which is used to include an image within an HTML document." [HTML Sourcebook at 3 (emphasis in original).] Finally, we emphasize that HTML is a *structured* language, which means that there are rules for where element tags can and cannot go. These rules are there to enforce an overall logical structure upon the document…. There are many [] structural [HTML] rules; they are given in detail in Chapter 2. This chapter illustrates the most obvious ones." [HTML Sourcebook at 3-4 (emphasis in original).] "You will notice that there is no ending tag </P> in Figure 1.1 to mark the end of the paragraphs. In HTML, the ending paragraph tag is option.. The rule is that a paragraph is *ended* by the next <P> tag starting another paragraph, or by any other tag that starts another *block* of text, such as a heading tag (<Hn>, or a quotation tag (<BLOCKQUOTE>), list tags, and so on." [HTML Sourcebook at 12 (emphasis in original).] "An HTML document is simply a text field in which certain strings of characters, called *tags*, mark regions of the document and assign special meanings to them. In the jargon of SGML, these regions are called *elements*. The tags are strings of characters surrounded by the less than (<) and greater than (>) signs." [HTML Sourcebook at 76 (emphasis in original).] "In some cases, the end tag is optional. This is the case when the end of an element can be unambiguously determined from the surrounding elements." [HTML Sourcebook at 77.] Web browsers (e.g., Mosaic, Netscape or Explorer) • contain methods to be used with HTML • HTML specifies "a rule set including subject matter specific tag pairs" (e.g. <A> & </A> and <BODY> & </BODY>) • HTML specifies "a separate tag pair for each of a plurality of different information types" (e.g. <A> & </A> for an 'anchor' information type) • HTML specifies a separate search rule for the <A> & </A> pair that directs the browser to check for partial URLs. "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML *hypertext anchors* and the URLs they contain . . . ." [HTML Sourcebook at xii.]

U.S. Patent No. 6,516,321	Disclosure

	
	
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`<BASE HREF="C:\www.stuff.edu/main/docs/file.html">`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**File.html**.]
	
	"Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF="stuff.html"> anchor text </A>`
	
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`
	
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html. Other equivalent partial URLs are then:
	/main/docs/stuff .html
	//www.stuff.edu/main/docs.html
	
	The former appropriates http:/www.stuff.edu from the current URL to complete the reference, while the latter appropriates only the http: part from the base URL of the current document.
	
	You can also use partial URLs to reference files in other directories; for example, from the example **file.html** the relative URL:
	../../main.html
	
	indicates the file **main.html** in the root HTTP directory, namely:
	http://www.stuff .edu/main.html
	
	Partial URLs are very useful when constructing large collections of documents that will be kept together. Of course relative URLs become invalid if a document is moved to a new directory or a new Internet site. This problem can be mitigated using the BASE element of the HTML, which is used to record the correct BASE URL of a document. If the document is moved, all relative URLs are determined relative to the URL recorded by the BASE element." [HTML Sourcebook at 167-68.]

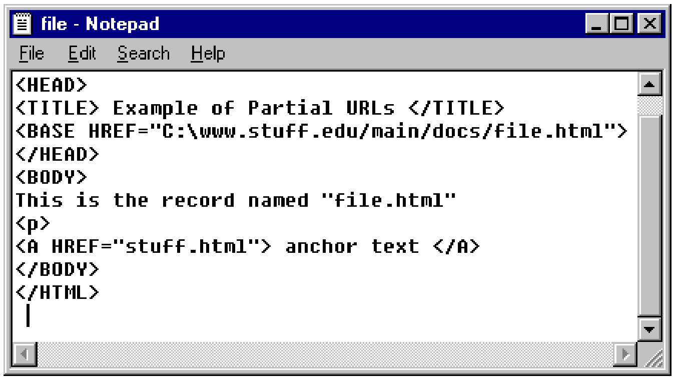
(a) receiving a record;	The HTML Sourcebook discloses receiving a document through a web browser: "You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers

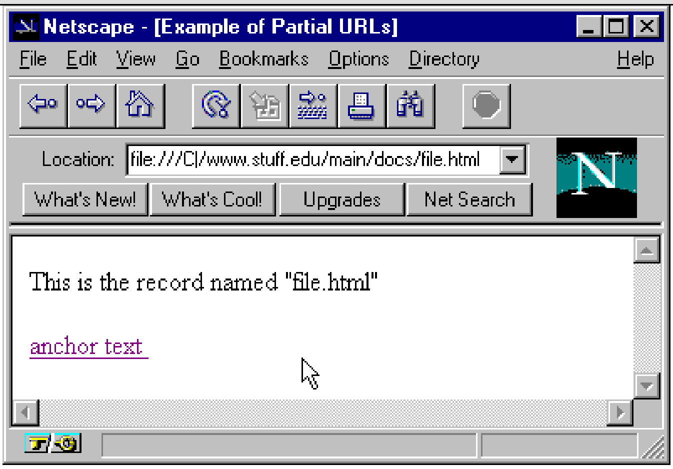
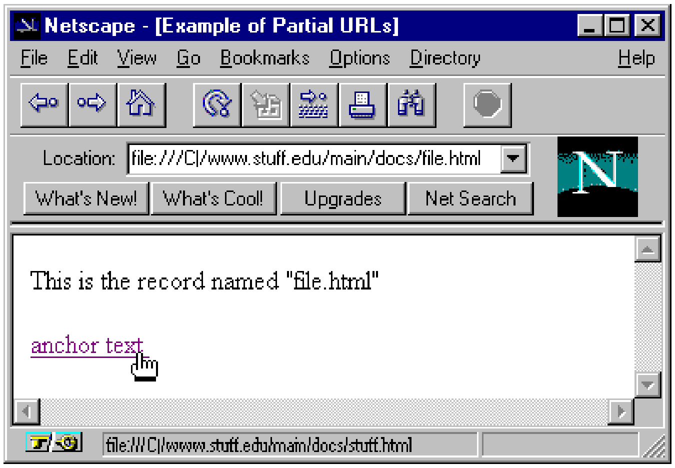
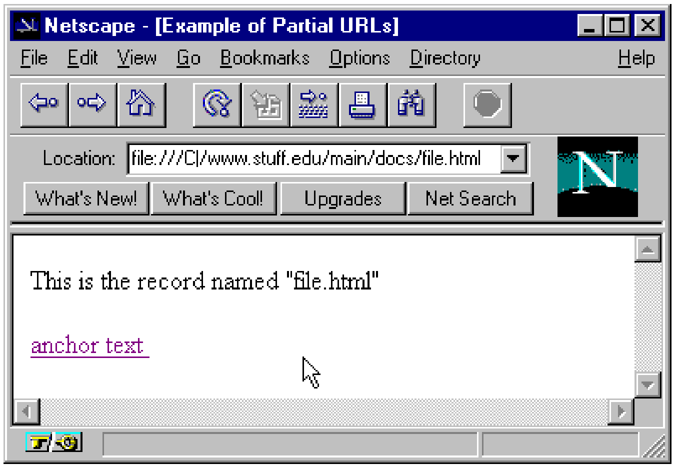
U.S. PATENT NO. 6,516,321	DISCLOSURE

	understand HTML *hypertext anchors* and the URLs they contain . . . .” [HTML Sourcebook at xii.]
	The file received (the “second record”) is the file entitled **file.html**:
	
	
	`<HEAD>`
	`<TITLE> Example of Partial URLs </TITLE>`
	`<BASE HREF="C:\www.stuff.edu/main/docs/file.html">`
	`</HEAD>`
	`<BODY>`
	`This is the record named "file.html"`
	`<p>`
	`<A HREF="stuff.html"> anchor text </A>`
	`</BODY>`
	`</HTML>`
	[**File.html**, viewed in Notepad.]
	“Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html”`
	[HTML Sourcebook at 167.]

(b) examining the record according to the search rules to identify record segments including information of each of the information types;	The HTML Sourcebook discloses a web browser parsing the text of a document, and based on structured HTML rules, determining which portions contain information of different types. [HTML Sourcebook 76-159.]
(c) when a record segment is identified which is of a particular information type: accessing the tag pair associated with the information type; inserting the begin tag before the identified segment and inserting the end tag after the identified segment.	When a web browser identifies different information types, HTML Sourcebook discloses inserting into the web page begin and end tags delineating elements recognized in HTML by a web browser. An example of the elements that would be recognized, and a description of their begin/end tags, can be found in the HTML Sourcebook at pages 76-159.

Claim 86. [i] A method for use with an application wherein specifying references (SRs) in one record to other records which are selectable to access the other records are visually distinguished from other record information so as to indicate selectability,	The HTML Sourcebook discloses a method to be used with an application, such as Mosaic, Cello or Netscape, where SRs that are capable of accessing other records are made selectable and visually distinguishable from other record information. For example:
	“You can write simple HTML documents and view them with a WWW browser, such as Mosaic, MacWeb, lynx, Cello, or Netscape . . . . Browsers understand HTML *hypertext anchors* and the URLs they contain . . . .” [HTML Sourcebook at xii.]
	“Suppose you originally access the document **file.html** using the full URL:
	`http://www.stuff.edu/main/docs/file.html`
	Within this document there is a hypertext reference containing a partial URL:
	`<A HREF=“stuff.html”> anchor text </A>`
	Where is this file? From inside **file.html**, any information not present in a URL reference is considered the *same* as that used to access the current document. Thus, the partial URL `stuff.html` is transformed into a full URL by appropriating the missing information from the URL used to access file.html. The completed URL is then:
	`http://www.stuff.edu/main/docs/stuff.html`

U.S. PATENT NO. 6,516,321	DISCLOSURE
	which indicates, as expected, that **stuff.html** is on the same server and in the same directory as file.html. Other equivalent partial URLs are then: /main/docs/stuff .html //www.stuff.edu/main/docs.html The former appropriates http:/www.stuff.edu from the current URL to complete the reference, while the latter appropriates only the http: part from the base URL of the current document. You can also use partial URLs to reference files in other directories; for example, from the example **file.html** the relative URL: ../../main.html indicates the file **main.html** in the root HTTP directory, namely: http://www.stuff .edu/main.html Partial URLs are very useful when constructing large collections of documents that will be kept together. Of course relative URLs become invalid if a document is moved to a new directory or a new Internet site. This problem can be mitigated using the BASE element of the HTML, which is used to record the correct BASE URL of a document. If the document is moved, all relative URLs are determined relative to the URL recorded by the BASE element." [HTML Sourcebook at 167-68]  <HEAD> <TITLE> Example of Partial URLs </TITLE> <BASE HREF="C:\www.stuff.edu/main/docs/file.html"> </HEAD> <BODY> This is the record named "file.html" <p> <A HREF="stuff.html"> anchor text </A> </BODY> </HTML> [**File.html**.] When a user viewing a document in a web browser such as is disclosed in the HTML Sourcebook hovers their mouse over the SR (e.g., the hypertext link anchor), the browser shows that the SR is selectable to access another record by displaying the fully resolved URL to the other record. As shown in the screenshots below, the SR (the "anchor text") is visually distinguished from the other information in the web page.

U.S. PATENT NO. 6,516,321	DISCLOSURE
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text [Navigator – Viewing **file.html**.]  Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.]
[ii] the method also for use with a system which enables a user to designate and also select SRs where designation comprises pointing to an SR without selection and,	The HTML Sourcebook discloses a method implemented by a system which allows a user to designate SRs by mousing over the SRs (anchor text) to determine more specific information about them (the URL), displayed in the lower left hand corner of the browser, without selection. See discussion above in connection with feature [i]. See also:  Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help What's New! What's Cool! Upgrades Net Search Location: file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text [Navigator – Viewing **file.html**.]

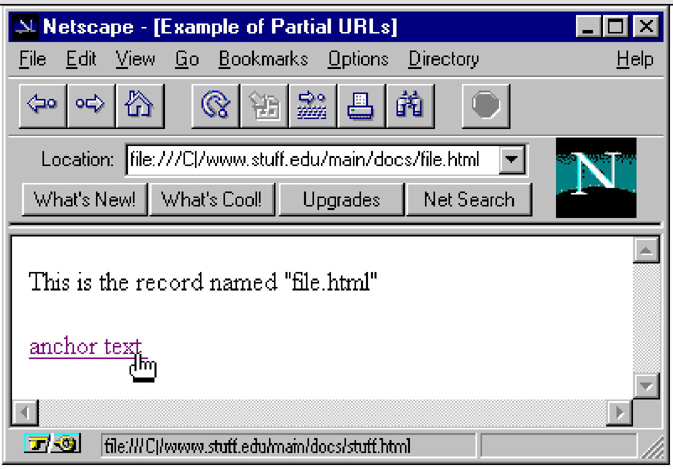
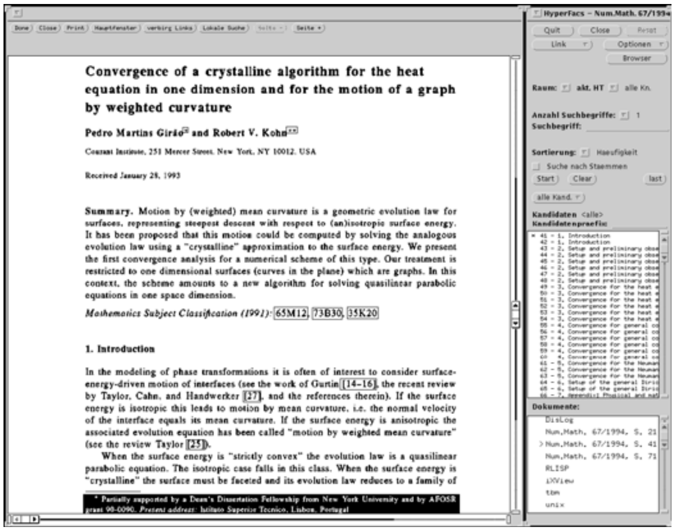
U.S. PATENT NO. 6,516,321	DISCLOSURE
	 Netscape - [Example of Partial URLs] File Edit View Go Bookmarks Options Directory Help Location: What's New! What's Cool! Upgrades Net Search file:///C|/www.stuff.edu/main/docs/file.html This is the record named "file.html" anchor text file:///C|/www.stuff.edu/main/docs/stuff.html [Navigator – Hovering cursor over hyperlink in **file.html**.]
[iii] wherein a seemingly general SR is modified by other record information which renders the SR relatively specific,	The HTML Sourcebook discloses a method implemented by a system to modify a general SR which renders the SR relatively specific, as discussed above in connection with features [i] and [ii]. The general SR is the anchor text. The other record information which renders the SR more specific is the BASE URL.
[iv] the method for indicating the specific nature of an SR prior to selection and comprising the steps of:	The HTML Sourcebook discloses a method implemented by system to indicate the specific nature of an SR prior to selection, as discussed in connection with the above feature [ii].
(a) when an SR is designated, indicating the specific nature of the SR	The HTML Sourcebook discloses a method implemented by system to indicate the specific nature of an SR once designated, as discussed in connection with the above feature [ii].

EXHIBIT 2c

APPENDIX C-4

SUPPLEMENTAL REPORT INVALIDITY ANALYSIS

Myka, A. & Güntzer, U., *Automatic Hypertext Conversion of Paper Document Collections*, in Selected Papers From the Digital Libraries Workshop on Digital Libraries: Current Issues, Lecture Notes In Computer Science, Vol. 916. Springer-Verlag, London, pp. 65-90 (N. R. Adam et al. eds., Springer-Verlag, London) (May 19-20, 1994) [hereinafter "Myka"]

U.S. PATENT NO. 6,516,321	DISCLOSURE
Claim 86. [i] A method for use with an application wherein specifying references (SRs) in one record to other records which are selectable to access the other records are visually distinguished from other record information so as to indicate selectability,	Myka discloses a digital library called the HyperFacs system that indicates specific information about the specifying reference when the user moves the cursor over the SR. Myka discloses the use of a pattern-matching technique to first identify candidate links and then to process additional information such as layout characteristics to further define the link. Recognized links (the SRs) are visually distinguished from other text by being surrounded by a box. Once the user's cursor moves over these boxes information is displayed to indicate selectability and the specific nature of the SR. HyperFacs displays both internal SRs (e.g., "*"), those that are highlighted within the document and refer to information about the target page, and external SRs (e.g., "65M12"), those that display possible actions to be performed with the link which are displayed in a central control window (right side of Figure 9). See, e.g.:  Done Close Print Hauptfenster verbing Links Lokale Suche Seite - Seite + HyperFacs – Num.Math. 67/1994 Quit Close Reset Link Optionen Browser Raum: akt. HT alle Kn. Anzahl Suchbegriffe: 1 Suchbegriff: Sortierung: Haeufigkeit Suche nach Staemmen Start Clear last alle Kand. Kandidaten <alle> Kandidatenpraefix: 41 = 1. Introduction 42 = 1. Introduction 43 = 2. Setup and preliminary obse 44 = 2. Setup and preliminary obse 45 = 2. Setup and preliminary obse 46 = 2. Setup and preliminary obse 47 = 2. Setup and preliminary obse 48 = 2. Setup and preliminary obse 49 = 3. Convergence for the heat e 50 = 3. Convergence for the heat e 51 = 3. Convergence for the heat e 52 = 3. Convergence for the heat e 53 = 3. Convergence for the heat e 54 = 3. Convergence for the heat e 55 = 4. Convergence for general co 56 = 4. Convergence for general co 57 = 4. Convergence for general co 58 = 4. Convergence for general co 59 = 4. Convergence for general co 60 4. Convergence for general co 61 – 5. Convergence for the Neuman 62 – 5. Convergence for the Neuman 63 – 5. Convergence for the Neuman 64 – 6. Setup of the general Diric 65 – 6. Setup of the general Diric 66 – 7. Appendix Physical and mat Dokumente: DisLog Num.Math. 67/1994. S. 21 Num.Math. 67/1994. S. 41 Num.Math. 67/1994. S. 71 RLISP iXView tbm unix **Convergence of a crystalline algorithm for the heat equation in one dimension and for the motion of a graph by weighted curvature** **Pedro Martins Girão[*] and Robert V. Kohn[**]** Courant Institute, 251 Mercer Street, New York, NY 10012, USA Received January 28, 1993 **Summary.** Motion by (weighted) mean curvature is a geometric evolution law for surfaces, representing steepest descent with respect to (an)isotropic surface energy. It has been proposed that this motion could be computed by solving the analogous evolution law using a "crystalline" approximation to the surface energy. We present the first convergence analysis for a numerical scheme of this type. Our treatment is restricted to one dimensional surfaces (curves in the plane) which are graphs. In this context, the scheme amounts to a new algorithm for solving quasilinear parabolic equations in one space dimension. *Mathematics Subject Classification (1991):* 65M12, 73B30, 35K20 **1. Introduction** In the modeling of phase transformations it is often of interest to consider surface-energy-driven motion of interfaces (see the work of Gurtin [14–16], the recent review by Taylor, Cahn, and Handwerker [27], and the references therein). If the surface energy is isotropic this leads to motion by mean curvature, i.e. the normal velocity of the interface equals its mean curvature. If the surface energy is anisotropic the associated evolution equation has been called "motion by weighted mean curvature" (see the review Taylor [25]). When the surface energy is "strictly convex" the evolution law is a quasilinear parabolic equation. The isotropic case falls in this class. When the surface energy is "crystalline" the surface must be faceted and its evolution law reduces to a family of * Partially supported by a Dean's Dissertation Fellowship from New York University and by AFOSR grant 90-0090. *Present address:* Instituto Superior Tecnico, Lisbon, Portugal [Figure 9: Screendump from a HYPERFACS session] "In order to be able to combine raster image representation with hypertext navigation, the common hypertext model had to be modified. With regard to the common model, the user is free to find his way through a document by means of choosing appropriate links (figure 7). Sometimes paths through the document are proposed by the system based on personal information needs and, or a user's profile. Within the modified model (fig. 8) a certain path through the document is always strengthened: the one that follows the linear

U.S. PATENT NO. 6,516,321	DISCLOSURE
	structure of the original document. Apart from this emphasis on a special path, the user is free to choose his own way through the document collection. Therefore, he may either navigate by means of browsing page-by-page or by means of traversing links. A screendump from a HyperFacs session is shown in figure 9. There, part of an article as published in the journal "Numerische Mathematik", Springer-Verlag, is presented. The viewer on the left side displays a page from an article containing eight different link sources and one highlighted link destination. The sources are marked by means of surrounding boxes which may be turned off by the user. The destination is marked by means of highlighting. In the given example the destination (footnote starting with '*') was reached and therefore highlighted by clicking on the framed asterisk ('*') following the first author's names. Additional information on a link (besides indicating the existence of a link by means of boxes) is presented to the user if he moves the cursor into the framed boxes: then, the type of information that is contained in the link destination is shown as well as the type of action that is triggered. On the right side of the figure the central control window is shown. There, different kind of actions may be initiated, e.g. full text searches, selection of objects or sets of objects, manual link generation and manual link deletion, generation and deletion of annotations, selection of the display type for showing search results, and indication whether internal nodes (sections) or external nodes (pages) should be used." [Myka at 86]
[ii] the method also for use with a system which enables a user to designate and also select SRs where designation comprises pointing to an SR without selection and,	Myka discloses a method to be used with the HyperFacs system which allows a user to designate without selection SRs if a user opts to do so, as discussed above in connection with feature [i]. See also: "Of course, it is not desirable to present each link and annotation to every user. This is true because of differences with regard to personal expertise or information needs. Therefore, access and modification rights with regard to links may be restricted to single users or groups of users. This restriction may be done either manually or by means of indicating appropriate access rights within the link type specification. Because the browser has been specifically tailored to the representation of raster images in connection with hypertext links, the hypertext functionality is completely integrated into the imaging system and works directly on top of the data base that is generated during the processing of library objects (see figure 1). Therefore, it can efficiently use the access structures within the data base. However, the usage of the browser again is restricted to a certain system environment, both with regard to hardware (SPARCstations) and software data base management system (TransBase)." [Myka at 86-89]
[iii] wherein a seemingly general SR is modified by other record information which renders the SR relatively specific,	The SRs are modified by other record information to make them more specific, as discussed above in connection with feature [i]. For example, an external SR is recognizes as such by other record information, such as layout characteristics, which enable the system to identify target document(s) and action(s) for that SR, all of which makes the SR relatively specific. See also: "How the evaluation of the processed document with regard to a link's parts (anchor and destination) is done, depends on the type of information: character pattern information (section 6.3.2) or layout information (section 6.3.2). The final generation of links depends on the fact whether linking is done locally (section 6.3.2) or globally (section 6.3.2)." [Myka at 78] "Furthermore, specific types of information are encoded using specific character patterns as clues. In printed documents, these cues may be based on

U.S. PATENT NO. 6,516,321	DISCLOSURE
	(additional) layout characteristics as well. The automatic detection of such cues within original electronic documents can be solved in a straightforward way by means of matching two fixed patterns exactly, thereby ignoring only typing errors. However, even locating the occurrences of character patterns only within an optically scanned document is more difficult. This is due to the fact that today's character recognition systems cannot provide complete for completely correct output [205]." [Myka at 79] "With regard to ordinary links, different pieces of layout information may be included. There, the analysis of layout information takes into consideration only those candidates that result from the filtering by means of information about character patterns." [Myka at 82]
[iv] the method for indicating the specific nature of an SR prior to selection and comprising the steps of:	Myka discloses a method implemented by the HyperFacs system which indicates the specific nature of an SR prior to selection, as discussed above in connection with features [i] – [iii].
(a) when an SR is designated, indicating the specific nature of the SR	Myka discloses a method implemented by the HyperFacs system which operates in a manner that once an SR is designated, the specific nature of that SR is revealed, as discussed above in connection with features [i] – [iii]. For example, the SR is an external link, mousing over it causes the nature of the SR to be displayed in the right hand window, when that specific nature was obtained as explained in feature [iii].

EXHIBIT 2d

APPENDIX C-5

SUPPLEMENTAL REPORT INVALIDITY ANALYSIS

Thistlewaite, Paul, *Automatic Construction and Management of Large Open Webs,* Info. Proc. & Mgmt., Vol. 33, No. 2, pp. 161-73 (Mar. 1997) [hereinafter, "PasTime"]

U.S. PATENT NO. 5,903,889	DISCLOSURE
Claim 1. A computer system with a plurality of data records on a plurality of databases, and a standardized format for addressing said data records, said computer system comprising:	PasTime discloses a computer system comprised of a multitude of databases, herein described as a Hyperbase comprised several sub-collections, that each contain plural data records and a standardized format for addressing records across the system. For example: "When a new file is detected the following automatic processes are applied: 1. the file is examined to determine which sub-collection it belongs to (e.g. Senate Hansard, Standing Orders, etc.) 2. a document identified index is generated listing the start byte location and bye extent of each atomic component document in the file together with canonical identifier for that component, such as 'Hasnard/Senate/1996/May/22/article_10'. . . " [Automatic Construction §5.1. A Case Study: A WWW Hyperbase for the Australian Parliament at 171.] *See also*: "The next step is to store the resulting HTML documents in a file system and, make them visible to the Web via a HTTP server running on their system (see Fig.1-HTML Embedded). . . However, for a document to be accessible over the Web it need not be stored, statically, in HTML format – the only requirement is that by the time the server provides the document to a client what it provides is HTML." [Automatic Construction §4. Representation, Storage and Installation of Hyperlinks at 166] "The approaches advocated in this paper have been used to build a hyperbase of the complete electronic document holdings of the Australian Parliament—in all, about 2 gigabytes of text. Those holdings include the Hansards for the House of Representatives and the Senate from 1981 onwards, Committee Transcripts and Reports, and Explanatory Memoranda for Bills. There is a range of other material: the agenda for Parliamentary meetings is provided by the Notice Papers, while the formal minutes of meetings are provided by the Votes and Proceedings (for the House of Representatives) and the Journals (for the Senate); the Standing Orders (for each House) provide the rules by which these meetings are conducted; the House Practice and Senate Practice books describe the formal and conventional rules and practices within the Parliament, and the Parliamentary Handbook provides other information, including biographical details of past and present Members and Senators.…. In all, there are approximately 250,000 separate atomic documents in the hyperbase[.]" [Automatic Construction §5. A Case Study: A WWW Hyperbase for the Australian Parliament at 170.] "There are two types of documents that a Web client browser can request: an index page for the collection, for some sub-collection, or for some compound document in a sub-collection; or an atomic document. Alternately, the Web client might issue a search request against the hyperbase, or some sub-collection within the hyperbase. All such requests are mediated by a suite of CGI programs on the server. In response to a request for a particular atomic document, the relevant CGI program will automatically perform the

U.S. Patent No. 5,903,889	Disclosure
	following tasks…." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "Fn.22: The server is an inexpensive Sun-5 workstation, with 32 megabytes of RAM and 10 gigabytes of disk." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "Parliament provides data in a number of formats, including a text database format, a word-processing format, and ASCII. These are, generally speaking, in large files that correspond to, say, a week or a Parliamentary session's worth of Hansard data, or the entire Parliamentary Handbook. Each file can be partitioned into smaller components of self-contained information, of a size more suitable for searching, browsing or delivering to the Web." [Automatic Construction §5. A Case Study: A WWW Hyperbase for the Australian Parliament at 170.] "[E]ach component document is indexed by a concordance-style IR subsystem, so that the hyperbase is searchable." [Automatic Construction §5.1. A Case Study: A WWW Hyperbase for the Australian Parliament at 171.] "The rapid growth of the World Wide Web has seen the emergence of some web sites that have hyperbases consisting of several hundred thousand web documents. Fn1: One such site is our own: http://pastime.anu.edu.au. For our purposes, a document can be viewed as an individually addressable Web page, accessible via a unique URL." [Automatic Construction §1. Introduction at 161.] "[F]or a document to be accessible over the Web it need not be stored, statically, in HTML format – the only requirement is that by the time the server provides the document to a client what it provides is HTML." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] "An additional benefit arising from representing links in Form (4) is that such links can be applied not only to documents residing on your server, but to any document you might retrieve from the Web." [Automatic Construction §4.3. Intensional Representation of Links at 168.]
(a) a user interface having an interactive display program for requesting one of said data records and displaying a plurality of interface supported data formats;	PasTime discloses an interactive display program, namely, a HTML browser, that is capable of requesting a plurality of data formats. For example: "There are two types of documents that a Web client browser can request: an index page for the collection, for some sub-collection, or for some compound document in a sub-collection; or an atomic document. Alternately, the Web client might issue a search request against the hyperbase, or some sub-collection within the hyperbase. All such requests are mediated by a suite of CGI programs on the server. In response to a request for a particular atomic document, the relevant CGI program will automatically perform the following tasks…." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "Parliament provides data in a number of formats, including a text database format, a word-processing format, and ASCII." [Automatic Construction §5. A Case Study: A WWW Hyperbase for the Australian Parliament at 170.] "There are two types of documents that a Web browser can request…."

U.S. PATENT NO. 5,903,889	DISCLOSURE
	[Automatic Construction §5.1 A Case Study: A WWW Hyperbase for the Australian Parliament at 171.]
(b) means for receiving a reference to a first data record from said interactive display program;	The corresponding structure for this element is the HTML browser. PasTime discloses a means for receiving a reference to a first data record from the interactive display program. For example: "All requests from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: (i) locate and load the original document from the server's file system…." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.]
(c) means for retrieving said first data record;	The corresponding structure for this element is the HTML browser in conjunction with the CGI program. Pastime discloses a means for retrieving a data record, as discussed above in connection with element (b). See also: "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it into a base HTML document containing HTML formatting tags[.]" [Automatic Construction §5.2. Client Requests and Server Responses at 171.]
(d) means for parsing said first data record to identify a reference to a second data record;	The corresponding structure for this element is the HTML browser in conjunction with the CGI program. PasTime discloses a means for parsing the first data record to identify a second data record. For example: "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: … 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added at the appropriate places to the base HTML document[.]" [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "As the objects of link relations are typically strings of characters in documents, lexical patterns are used to define the predicates for representing link anchors. Patterns can be as simple as literal strings, or complex regular expressions. A source (or target) anchor is located in the source (or target) document not by specifying the offset and extent of the source (or target) expression, but rather by specifying the pattern that the expression must match." [Automatic Construction §4.3. Intensional Representation of Links at 167.] "Consequently, at the time of serving a document to a client, a link is only embedded into the byte stream by the CGI program doing the pattern detection and link resolution if (i) an expression in the document matches one of the source anchor patterns applicable to that document, and (ii) the corresponding document exists." [Automatic Construction §4.3. Intensional Representation of Links at 168.] "To illustrate, take an example from the documents of the Australian Parliament. The documents which record the debates in the parliament are

U.S. PATENT NO. 5,903,889	DISCLOSURE
	called the 'Hansards'. Another document, which lists the rules under which such debates are conducted, is called the 'Standing Orders.' … Can we devise a pattern for linking any reference to the Standing Orders or a particular standing order in a Hansard document to the Standing Orders document itself? Will the following regular expression, *P*, be sound and complete: "standing" (SPACE) + "order" ((SPACE) + (0-9) + (A-Z)?)? *P* will match any string consisting of the word 'standing' followed by one or more spaces (including newline characters) followed the word 'order', optionally followed by more space and a string of digits, optionally followed by an uppercase alphabetical character." [Automatic Construction §4.4. Devising Sound and Complete Patterns for Referential Links at 169.]
(e) means for modifying said reference to said second data record to create an address, said address being operable to retrieve said second data record; and	The structure corresponding to this element is the HTML browser in conjunction with the CGI program. PasTime discloses a means for modifying a reference to a second data record in order to create an address which is operable to retrieve the second data record. For example: "The method most familiar to authors using HTML to create a hyperbase for the Web is to take the original documents, add HTML markers for formatting the documents presentationally, and add link information into documents which contain link sources. (Fn16: If the target of a link is not a whole document but rather some point in a document, then additional markup (a named tag) must be added to the target document.) The next step is to store the resulting HTML documents in a file system, and make them visible to the Web via a HTTP server running o their system (see Fig. 1—HTML Embedded)." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] "Consequently, at the time of serving a document to a client, a link is only embedded into the byte stream by the CGI program doing the pattern detection and link resolution if (i) an expression in the document matches one of the source anchor patterns applicable to that document, and (ii) the corresponding document exists." [Automatic Construction §4.3. Intensional Representation of Links at 168.] "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it into a base HTML document containing HTML formatting tags…. 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added a the appropriate places to the base HTML document 3. finally, the link detection and instantiation routines are employed to embed any referential links into the base HTML document, and the resulting byte stream is dispatched to the client." [Automatic Construction §5.2. Client Requests and Server Responses at 171.]
(f) means for sending said modified first data record to said interactive display program.	The structure corresponding to this element is the HTML browser in conjunction with the CGI program. Pastime discloses a means for sending the modified first data record to the display program, as discussed above in connection with element (e). See also:

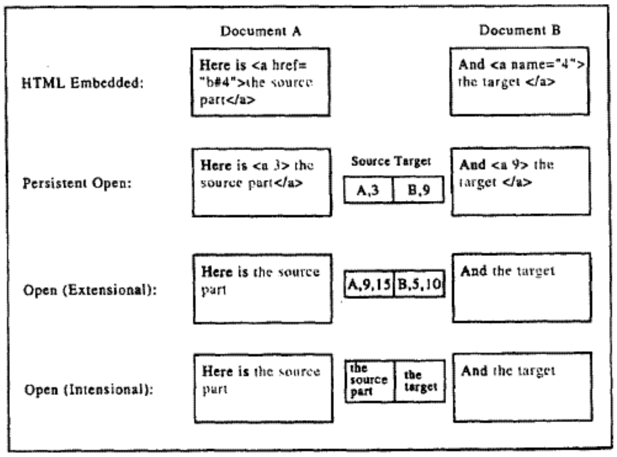
U.S. PATENT NO. 5,903,889	DISCLOSURE
	"Referential links can be displayed using the standard technique of underlining or otherwise highlighting the referring expression." [Automatic Construction §3.2. Referential links at 164.]
Claim 7. The computer system of claim 1, wherein said reference to said second data record comprises a keyword phrase.	PasTime discloses a computer system where a reference to a second data record comprises a keyword phrase. See also the above discussion for claim 1. See also: "In the case of referential links, the string that matches the source anchor expression is often a strong indicator of the name of the target document. Significant economies of representation and easier link maintenance can be achieved if the source anchor pattern for a referential link can be generalized, and the identifier for the appropriate target document can be computed as a function of the string that matches the source anchor pattern. To illustrate, I may have devised separate patterns that match the names of all pieces of legislation discussed in the Australian Parliament, for example: one that matches "Native Title Bill 1992" and links such expressions to the document named Native_Title_Bill_1992, and one that matches "Taxation Amendment Bill 1995" and links such expression to the document named Taxation_Amendment_Bill_1995, and so on. However, if a pattern can be devised that will match *any* reference to *any* piece of legislation, then in such a case it is an easy matter to determine the correct target document identifier from the string that matches the source anchor expression (Form (4))." $\{\text{source} = <, S - \text{pattern}>, \text{target} = <\text{func}(\text{match\_}S - \text{pattern}), T - \text{pattern}>] \qquad (4)$ [Automatic Construction §4.3. Intensional Representation of Links at 168.]

U.S. PATENT NO. 6,516,321	DISCLOSURE
Claim 1. A method for identifying a referenced record referenced in a referencing record wherein the referenced record is referenced in the referencing record by at least a combination including a data reference (DR) and a modifier reference (MR), the method comprising the steps of:	PasTime discloses a method implemented by a system to identify a second record, referenced in a first record which is identified by at least a DR and an MR. For example: "It is a mistake to link a reference to a Member, say their name, in a 1985 Hansard to their current biographical details in the current Parliamentary Handbook, which may now describe him or her as 'the Minister for Transport' when at the time he or she was not. Politically speaking, it is an even worse mistake to link the referring expression 'the Prime Minister' occurring in the same 1985 Hansard to biographical details of the current incumbent. This problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4 Devising Referential Link Patterns at 172.] "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: … 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added at the appropriate places to the base HTML document[.]" 3. finally, the link detection and instantiation routines are employed to embed any referential links into the base HTML document, and the resulting byte stream is dispatched to the client." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "The problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4. Devising Referential Link Patterns at 172.] "For example, the first equivalence operation may involve the case of the expression – e.g. those sentences (noun phrases) containing 'Bill' of 'BILL' as opposes to 'bill.' "Later functions could involve checking for the presence of a year designator (e.g. "…Bill … 1993") and so on. At each stage, examine each equivalence class to determine whether the candidate referring expressions are sound." [Automatic Construction §4.4 Devising Sound and Complete Patterns for Referential Links at 170
(i) receiving the referencing record;	PasTime discloses a method implemented by a system to receive the referencing record. For example: "All requests from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: (i) locate and load the original document from the server's file system…." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] See also the above discussion in connection with the preamble.
(ii) analyzing the referencing record to	PasTime disclose analyzing the referencing record to identify a DR, once it is

U.S. PATENT NO. 6,516,321	DISCLOSURE
identify a DR, when a DR is identified:	identified. Examples of DRs are "Prime Minister" and "Bill." "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: … 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added at the appropriate places to the base HTML document[.]" [Automatic Construction §5.2. Client Requests and Server Responses at 171] "As the objects of link relations are typically strings of characters in documents, lexical patterns are used to define the predicates for representing link anchors. Patterns can be as simple as literal strings, or complex regular expressions. A source (or target) anchor is located in the source (or target) document not by specifying the offset and extent of the source (or target) expression, but rather by specifying the pattern that the expression must match." [Automatic Construction §4.3. Intensional Representation of Links at 167] "Consequently, at the time of serving a document to a client, a link is only embedded into the byte stream by the CGI program doing the pattern detection and link resolution if (i) an expression in the document matches one of the source anchor patterns applicable to that document, and (ii) the corresponding document exists." [Automatic Construction §4.3. Intensional Representation of Links at 168] "To illustrate, take an example from the documents of the Australian Parliament. The documents which record the debates in the parliament are called the 'Hansards'. Another document, which lists the rules under which such debates are conducted, is called the 'Standing Orders.' … Can we devise a pattern for linking any reference to the Standing Orders or a particular standing order in a Hansard document to the Standing Orders document itself? Will the following regular expression, *P*, be sound and complete: "standing" (SPACE) + "order" ((SPACE) + (0-9) + (A-Z)?)? *P* will match any string consisting of the word 'standing' followed by one or more spaces (including newline characters) followed the word 'order', optionally followed by more space and a string of digits, optionally followed by an uppercase alphabetical character." [Automatic Construction §4.4. Devising Sound and Complete Patterns for Referential Links at 169]
(a) identifying an MR rule set (MRRS) specifying the relationship between an MR and the DR;	PasTime discloses the feature of identifying an MR rule set (MRRS) that specifies the relationship between an MR and the DR. For example, the MRRS for "Prime Minister" is to ascertain the date of the document. The MRRS for "Bill" is to look for date information in the surrounding text. "As was noted earlier, links are relations. Relations can be represented extensionally, as the set of ordered n-tuples that satisfy the relation—for example, the sibling relation xSy can be represented by listing all the tuples of brothers and sisters: <John,Peter>, <John,Mary>, and so on. Alternatively, a relation can be represented intensionally, as the predicate that determines the relation—for example, "sibling" can be represented as "has a common parent" (i.e. xSy iff Px=Py)." [Automatic Construction §4.3 Intensional Representation of Links at 167] "As the objects of link relations are typically strings of characters in documents, lexical patterns are used to define the predicates for representing

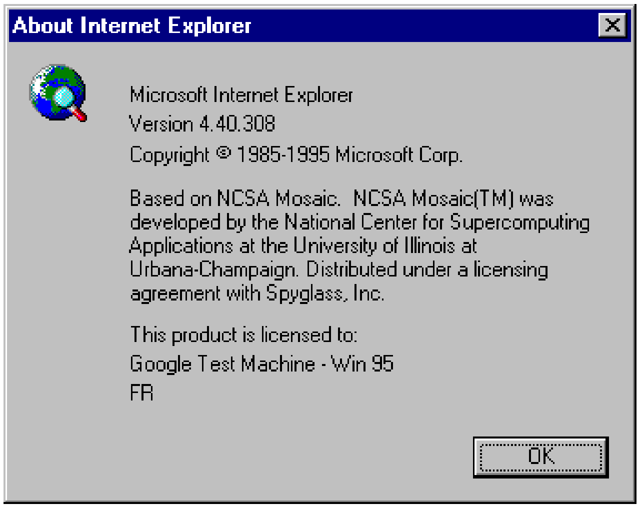
U.S. PATENT NO. 6,516,321	DISCLOSURE
	link anchors. Patterns can be as simple as literal strings, or complex regular expressions." [Automatic Construction §4.3. Intensional Representation of Links at 167] "It is a mistake to link a reference to a Member, say their name, in a 1985 Hansard to their current biographical details in the current Parliamentary Handbook, which may now describe him or her as 'the Minister for Transport' when at the time he or she was not. Politically speaking, it is an even worse mistake to link the referring expression 'the Prime Minister' occurring in the same 1985 Hansard to biographical details of the current incumbent. This problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4 Devising Referential Link Patterns at 172] "Later functions could involve checking for the presence of a year designator (e.g. "…Bill … 1993") and so on. At each stage, examine each equivalence class to determine whether the candidate referring expressions are sound." [Automatic Construction §4.4 Devising Sound and Complete Patterns for Referential Links at 170) (emphasis added).] "The problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4. Devising Referential Link Patterns at 172]
(b) analyzing the referencing record in accordance with the MRRS to identify the existence of the MR and, when the MR is identified;	PasTime discloses analyzing the second record in accordance with an MR rule set in order to identify the existence of an MR, as discussed above in connection with element [ii]. In the example of the DR "Prime Minister," the MR is the date. In the example of the DR "Bill," the MR is also the date. See also: "3. partition: using clustering techniques, or any function that gives rise to appropriate equivalence classes, partition the set of retrieved sentences (noun phrases) into equivalence classes. This is often an iterative operation, with successive refinements to the equivalence function. For example, the first equivalence operation may involve the case of the expression—e.g. those sentences (noun phrases) containing 'Bill' or 'BILL' as opposed to 'bill'. Later functions could involve checking for the presence of a year designator (e.g. "…Bill … 1993") and so on. At each stage, examine each equivalence class to determine whether the candidate referring expressions are sound." [Automatic Construction §4.4 Devising Sound and Complete Patterns for Referential Links at 170] "The problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4. Devising Referential Link Patterns at 172]
(c) identifying the referenced	PasTime discloses identifying the referenced record associated with the

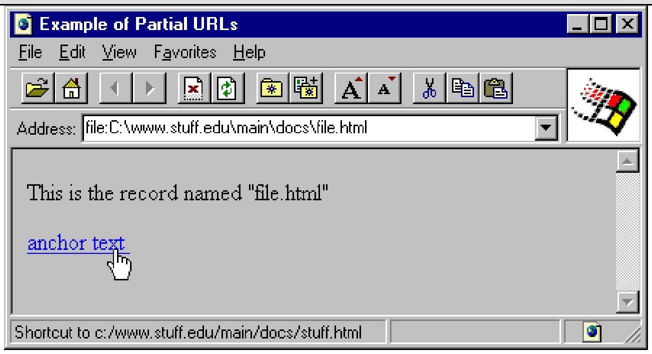
U.S. PATENT NO. 6,516,321	DISCLOSURE
record associated with the DR/MR combination.	DR/MR combination. For example: "All requests from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: (i) locate and load the original document from the server's file system…." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it into a base HTML document containing HTML formatting tags…. 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added a the appropriate places to the base HTML document 3. finally, the link detection and instantiation routines are employed to embed any referential links into the base HTML document, and the resulting byte stream is dispatched to the client." [Automatic Construction §5.2. Client Requests and Server Responses at 171] "The problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4. Devising Referential Link Patterns at 172]
Claim 24. The method of claim 1 further including the step of linking the record reference to the referenced record.	PasTime discloses the method implemented by a system that includes the step of linking the reference to the referenced record. For example: "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it into a base HTML document containing HTML formatting tags…. 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added a the appropriate places to the base HTML document 3. finally, the link detection and instantiation routines are employed to embed any referential links into the base HTML document, and the resulting byte stream is dispatched to the client." [Automatic Construction §5.2. Client Requests and Server Responses at 171] "Consequently, at the time of serving a document to a client, a link is only embedded into the byte stream by the CGI program doing the pattern detection and link resolution if (i) an expression in the document matches one of the source anchor patterns applicable to that document, and (ii) the corresponding document exists." [Automatic Construction §4.3. Intensional Representation of Links at 168]
Claim 27. [i] A method to be used with a rule set including subject matter specific tag pairs and corresponding search rules,	PasTime discloses a system that implements a method that uses rules to add subject matter specific tag pairs to a record, applying HyperPhrase's application of this claim. For example: "The method most familiar to authors using HTML to create a hyperbase for the Web is to take the original documents, add HTML markers for formatting the documents presentationally, and add link information into documents

U.S. PATENT NO. 6,516,321	DISCLOSURE
	which contain link sources. Fn16: If the target of a link is not a whole document but rather some point in a document, then additional markup (a named tag) must be added to the target document." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] "In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it to a base HTML document containing HTML formatting tags (this conversion is done using routines specifically developed for each collection)." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "A request from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: … (ii) add appropriate HTML markers to achieve the desired presentational format for the client's browser[.]" [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166 .] "In the persistent open strategy, a separate link database is kept in which the source and target information for a link are represented as a couple, consisting of two anchors, each of which records the document in which the anchor resides and the identifier of the tag in the document which indicates where the anchor is to be sited." [Automatic Construction §4.1 Open Link Strategies at 166-67.]  Document A Document B HTML Embedded: Here is <a href= "b#4">the source part</a> And <a name="4"> the target </a> Persistent Open: Here is <a 3> the source part</a> Source Target A,3 B,9 And <a 9> the target </a> Open (Extensional): Here is the source part A,9,15 B,5,10 And the target Open (Intensional): Here is the source part the source part the target And the target Fig. 1. Alternative approaches to link representation and storage. This reference also discloses: "Detection of macroscopic structural links relies on either having an explicit representation of the document in each collection and subcollection, or on deriving such a representation from the hierarchical file system directory structure in which the documents are stored. Detection of microscopic structural links usually relies on being able to unpack and individual documents internal structure, either using existing markup associated with the document (e.g. SGML DTDs) or using the extant techniques for table-of-content creation." [Automatic Construction §3.1 at 164.] PasTime discloses use of a web browser. A person of skill in the art would have known, prior to the filing of the patent, that popular web browsers such as Internet Explorer 1.0 and Netscape Navigator 1.22 both disclosed the features of this claim.

U.S. PATENT NO. 6,516,321	DISCLOSURE
	Specifically, A person of skill in the art would understand that • web browsers (e.g., Netscape or Explorer) contain methods to be used with HTML • HTML specifies "a rule set including subject matter specific tag pairs" (e.g. <A> & </A> and <BODY> & </BODY>) • HTML specifies "a separate tag pair for each of a plurality of different information types" (e.g. <A> & </A> for an 'anchor' information type) • HTML specifies a separate search rule for the <A> & </A> pair that directs the browser to check for partial URLs. "There are two types of documents that a Web client browser can request: an index page for the collection, for some sub-collection, or for some compound document in a sub-collection; or an atomic document. Alternately, the Web client might issue a search request against the hyperbase, or some sub-collection within the hyperbase. All such requests are mediated by a suite of CGI programs on the server. In response to a request for a particular atomic document, the relevant CGI program will automatically perform the following tasks…." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "The rapid growth of the World Wide Web has seen the emergence of some web sites that have hyperbases consisting of several hundred thousand web documents. Fn1: One such site is our own: http://pastime.anu.edu.au. For our purposes, a document can be viewed as an individually addressable Web page, accessible via a unique URL." [Automatic Construction §1. Introduction at 161.] "Parliament provides data in a number of formats, including a text database format, a word-processing format, and ASCII. These are, generally speaking, in large files that correspond to, say, a week or a Parliamentary session's worth of Hansard data, or the entire Parliamentary Handbook. Each file can be partitioned into smaller components of self-contained information, of a size more suitable for searching, browsing or delivering to the Web." [Automatic Construction §5. A Case Study: A WWW Hyperbase for the Australian Parliament at 170.]
[ii] a separate tag pair for each of a plurality of different information types	PasTime discloses separate tag pairs for each of a multitude of different information types, as discussed above in connection with feature [i].
[iii] and a separate search rule for each pair,	PasTime discloses separate search rules for subject matter specific pair, as discussing above in connection with feature [i].
[iv] each pair including a begin tag and an end tag, the method comprising the steps of:	PasTime discloses begin and end tags for each subject matter specific pair, as discussing above in connection with feature [i].
(a) receiving a record**;**	PasTime discloses the method implemented by a system to receive a record, as discussed above in connection with feature [i]. For example: "There are two types of documents that a Web client browser can request: an index page for the collection, for some sub-collection, or for some compound document in a sub-collection; or an atomic document. Alternately, the Web client might issue a search request against the hyperbase, or some sub-collection within the hyperbase. All such requests are mediated by a suite of CGI programs on the server. In response to a request for a particular atomic document, the relevant CGI program will automatically perform the following tasks…." [Automatic Construction §5.2. Client Requests and Server Responses at 171]

U.S. Patent No. 6,516,321	Disclosure
	"The rapid growth of the World Wide Web has seen the emergence of some web sites that have hyperbases consisting of several hundred thousand web documents. Fn1: One such site is our own: http://pastime.anu.edu.au. For our purposes, a document can be viewed as an individually addressable Web page, accessible via a unique URL." [Automatic Construction §1. Introduction at 161] A web page is received when a user opens it in either Internet Explorer or Netscape Navigator. "All requests from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: (i) locate and load the original document from the server's file system…." [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166.] "In response to a request for a particular index page, the relevant CGI program will dynamically generate a HTML page containing the HTML page containing the index information, by examining the appropriate document identifier index and attribute information…. In response to a request for a particular atomic document, the relevant CGI program will automatically undertake the following tasks: 1. extract the component document from the original file, and depending on the sub-collection that it belongs to, convert it into a base HTML document containing HTML formatting tags…. 2. the attribute information for the document, and the document identifier index, are then examined to compute the structural links for this document, and these are then added a the appropriate places to the base HTML document 3. finally, the link detection and instantiation routines are employed to embed any referential links into the base HTML document, and the resulting byte stream is dispatched to the client." [Automatic Construction §5.2. Client Requests and Server Responses at 171.]
(b) examining the record according to the search rules to identify record segments including information of each of the information types;	PasTime discloses a method to implemented by a system to examine a record according to search rules in order to identify record segments, as discussed above in connection with feature [i] and element [a]. A person of skill in the art understands that when Internet Explorer or Netscape Navigator opens a web page, the web browser parses the page to see if any HTML elements are present, as the browser renders the web page for display to the user. Such browser will identify different types of information, such as anchor tags, headers, etc… By way of further example, a listing of many of the types of information looked for can be found in the HTML Sourcebook by Ian S. Graham (1995), which a person of ordinary skill in the art would know.
(c) when a record segment is identified which is of a particular information type: accessing the tag pair associated with the information type; inserting the begin tag before the identified segment and inserting the end tag after the identified segment.	PasTime discloses begin and end tag insertion after a record segment has been identified, as discussed above in feature [i] and element (a). See also: "A request from a client, in the form of a URL, can specify which original document is being sought, and the CGI program indicated in the URL can then: … (ii) add appropriate HTML markers to achieve the desired presentational format for the client's browser[.]" [Automatic Construction §4. Representation, Storage and Instantiation of Hyperlinks at 166 .] A person of skill in the art understands that when particular types of information are recognized by the web browser as it renders the page for

U.S. PATENT NO. 6,516,321	DISCLOSURE
	viewing, the browser inserts begin and end tags specific to that information type into the web page being displayed to the user.
Claim 86. [i] A method for use with an application wherein specifying references (SRs) in one record to other records which are selectable to access the other records are visually distinguished from other record information so as to indicate selectability,	PasTime discloses a method implemented by an application that indicates specific information about the specifying reference when the user moves the cursor over the SR, making the SR visually distinguishable from other record information so as to indicate selectability. For example: "The problem was corrected by extending Form (4) link descriptors to take additional arguments to the function for calculating the target of a source anchor expression—in addition to taking whatever string matched the pattern, the function can also take attribute information for the document (in this case, the date), which enables the identification of the correct target to be computed." [Automatic Construction §5.4. Devising Referential Link Patterns at 172.] To the extent not expressly disclosed, a person of ordinary skill in the art PasTime discloses use of a web browser. A person of skill in the art would have known, prior to the filing of the patent, that popular web browsers such as Internet Explorer 1.0 and Netscape Navigator 1.22 both disclosed the features of this claim. Specifically, when a user of Internet Explorer or Netscape Navigator opens a web page and hovers the mouse over the hypertext link anchor that is displayed by the browser in underline on the rendered page, without clicking the mouse, the browser indicates the specific nature of the URL by displaying the location and name of the document associated with the hypertext link anchor in the bottom portion of the web browser window. Moreover, prior to displaying the rendered web page to the user, both Internet Explorer and Netscape navigator had the ability to resolve partial URLs, searching the web page for a partial URL that referred to a document and either a BASE HREF element or the URL used to access the web page being opened, whereby the browser would glean from either the BASE HREF or the URL used to access the web page being viewed the base or root URL necessary to resolve the partial URL into a full URL, rendering specific the target document referenced by the partial URL.  Microsoft Internet Explorer Version 4.40.308 Copyright © 1985-1995 Microsoft Corp. Based on NCSA Mosaic. NCSA Mosaic(TM) was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Distributed under a licensing agreement with Spyglass, Inc. This product is licensed to: Google Test Machine - Win 95 FR

U.S. PATENT NO. 6,516,321	DISCLOSURE
	 Example of Partial URLs File Edit View Favorites Help Address: file:C:\www.stuff.edu\main\docs\file.html This is the record named "file.html" anchor text Shortcut to c:/www.stuff.edu/main/docs/stuff.html "There are two types of documents that a Web client browser can request: an index page for the collection, for some sub-collection, or for some compound document in a sub-collection; or an atomic document. Alternately, the Web client might issue a search request against the hyperbase, or some sub-collection within the hyperbase. All such requests are mediated by a suite of CGI programs on the server. In response to a request for a particular atomic document, the relevant CGI program will automatically perform the following tasks…." [Automatic Construction §5.2. Client Requests and Server Responses at 171.] "The rapid growth of the World Wide Web has seen the emergence of some web sites that have hyperbases consisting of several hundred thousand web documents. Fn1: One such site is our own: http://pastime.anu.edu.au. For our purposes, a document can be viewed as an individually addressable Web page, accessible via a unique URL." [Automatic Construction §1. Introduction at 161.] "Parliament provides data in a number of formats, including a text database format, a word-processing format, and ASCII. These are, generally speaking, in large files that correspond to, say, a week or a Parliamentary session's worth of Hansard data, or the entire Parliamentary Handbook. Each file can be partitioned into smaller components of self-contained information, of a size more suitable for searching, browsing or delivering to the Web." [Automatic Construction §5. A Case Study: A WWW Hyperbase for the Australian Parliament at 170.]
[ii] the method also for use with a system which enables a user to designate and also select SRs where designation comprises pointing to an SR without selection and,	PasTime discloses a method implemented by a system which enables a user to designate and also select SRs where the designation identifies an SR without selection, as discussed above in connection with feature [i].
[iii] wherein a seemingly general SR is modified by other record information which renders the SR relatively specific,	PasTime discloses a method implemented by a system where a general SE is modified by other record information to become more specific, as discussed above in connection with feature [i].
[iv] the method for indicating the specific nature of an SR prior to selection and comprising the steps of:	PasTime discloses a method implemented by a system where the specific nature of an SR prior to selection is indicated, as discussed above in connection with feature [i].
(a) when an SR is designated, indicating the specific nature of the SR	PasTime discloses a method implemented by a system where an SR is designated indicating the specific nature of the SR, as discussed above in connection with feature [i]. See also:

U.S. PATENT NO. 6,516,321	DISCLOSURE
	"Referential links can be displayed using the standard technique of underlining or otherwise highlighting the referring expression." [Automatic Construction §3.2. Referential links at 164.]

EXHIBIT 2e

APPENDIX C-7

SUPPLEMENTAL REPORT INVALIDITY ANALYSIS

MUC-6 Named Entity Task Definition Task Overview (Version 2.0, 31 May 95); (see, for example, http://cs.nyu.edu/faculty/grishman/NEtask20.book_3.html#HEADING4)(June 2, 1995): MUC-6, 6th Message Understanding Conference (Nov. 6-8 1995) [hereinafter, Task Overview"]

U.S. PATENT NO. 6,516,321	DISCLOSURE

Claim 27. [i] A method to be used with a rule set including subject matter specific tag pairs and corresponding search rules,	Task Overview discloses a system that implements a method to add subject matter specific tags, such as PERSON, LOCATION and DATE, according to corresponding search rules. For example, the entirety of the MUC-6 TASK OVERVIEW:
	2 TASK OVERVIEW
	2.1 - Markup Description
	2.2 - Named Entities (ENAMEX tag element)
	2.3 - Temporal Expressions (TIMEX tag element)
	2.4 - Number Expressions (NUMEX tag element)
	2.1 Markup Description
	The output of the systems to be evaluated will be in the form of SGML text markup. The only insertions allowed during tagging are tags enclosed in angled brackets. No extra whitespace or carriage returns are to be inserted; otherwise, the offset count would change, which would adversely affect scoring.
	The markup will have the following form:
	`<ELEMENT-NAME ATTR-NAME="ATTR-VALUE" ...>text-string</ELEMENT-NAME>`
	Example:
	`<ENAMEX TYPE="ORGANIZATION">Taga Co.</ENAMEX>`
	The markup is defined in SGML Document Type Descriptions (DTDs), written for MUC-6 use by personnel at MITRE and maintained by personnel at NRaD. The DTDs enable annotators and system developers to use SGML validation tools to check the correctness of the SGML-tagged texts produced by the annotator or the system. The validation tools are available to MUC-6 participants in the file called muc6-sgml-tools. Annotators are using a software tool provided for MUC-6 by SRA Corporation to assist in generating the answer keys to be used for system training and testing.
	2.2 Named Entities (ENAMEX tag element)
	This subtask is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via the TYPE attribute as follows:
	ORGANIZATION: named corporate, governmental, or other organizational

U.S. PATENT NO. 6,516,321	DISCLOSURE
	entity PERSON: named person or family LOCATION: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.) **2.3 Temporal Expressions (TIMEX tag element)** This subtask is for "absolute" temporal expressions only; explanation is provided in appendix B. The tagged tokens are categorized via the TYPE attribute as follows: DATE: complete or partial date expression TIME: complete or partial expression of time of day **2.4 Number Expressions (NUMEX tag element)** This subtask is for two useful types of numeric expressions, monetary expressions and percentages. The numbers may be expressed in either numeric or alphabetic form. The task covers the complete expression, which is categorized via the TYPE attribute as follows: MONEY: monetary expression PERCENT: percentage
[ii] a separate tag pair for each of a plurality of different information types	Task Overview discloses a separate tag pair for different information types, as discussed above in connection with feature [i]. For example, the name entity tag element tags, Organization, Person and Location. See also: "2.2 Named Entities (ENAMEX tag element) This subtask is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via the TYPE attribute as follows: ORGANIZATION: named corporate, governmental, or other organizational entity PERSON: named person or family LOCATION: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)"
[iii] and a separate search rule for each pair,	Task Overview discloses a separate search rule for each pair, as discussed above in connection with feature [i]. For example, the Name Entities tag element ENAMEX, is governed by the following rules: "This subtask is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via the TYPE attribute . . ."
[iv] each pair including a begin tag and an end tag, the method comprising the steps of:	Task overview discloses a subject matter specific tags including both begin and end tags, as discussed above in connection with feature [i]. See also: "Example: <ENAMEX TYPE="ORGANIZATION">Taga Co.</ENAMEX>"
(a) receiving a record**;**	Task Overview discloses a system that implements a method to receive a record for subsequent processing and tagging, as discussed above in feature [i].
(b) examining the record according to the search rules to identify record segments	Task Overview discloses a system that implements a method to examine a record according to search rules to identify record segments, including

U.S. PATENT NO. 6,516,321	DISCLOSURE
including information of each of the information types;	information types, as discussed above in feature [i].
(c) when a record segment is identified which is of a particular information type: accessing the tag pair associated with the information type; inserting the begin tag before the identified segment and inserting the end tag after the identified segment.	Task Overview discloses a system that implements a method to insert begin and end tags before and after an identified record segment, as discussed above in connection with feature [i].

EXHIBIT 2f

Appendix C-8

Supplemental Report Invalidity Analysis

Aberdeen, J. et. al., *MITRE: Description of the Alembic System Used for MUC-6,* Proc. of the 6th Conf. on Message Understanding, pp. 141-155, Columbia, Maryland (Nov. 6-8 1995) [hereinafter "Aberdeen"]

U.S. Patent No. 6,516,321	Disclosure
Claim 27. [i] A method to be used with a rule set including subject matter specific tag pairs and corresponding search rules,	Aberdeen discloses a system that implements an automatic method that uses rules to add subject matter specific tag pairs to a record, such as a person's name and their title (e.g., <ttl>Mr.</ttl><person>James</person>). Aberdeen contains extensive disclosure regarding the rules that are used to identify record segments that contain information of various types. For example: "Rule sequences now underlie all the major processing steps in Alembic: part-of-speech tagging, syntactic analysis, inference, and even some of the set-fill processing in the Template Element task (TE). . . . The rules acquired in this way also have the characteristic that they allow one to readily mix hand-crafted and machine-learned elements." [Aberdeen at 141.] "The central innovation in the system is its approach to syntactic analysis, which is now performed through a sequence of phrase-finding rules that are processed by a simple interpreter. The interpreter has somewhat less recognition power than a finite-state machine, and operates by successively relabeling the input according to the rule actions. . . In support of the syntactic phrase finder, or phraser as we call it, the input text must be tagged for part-of-speech. This part-of-speech tagging is the principal role of the UNIX preprocess, and it is itself supported by a number of pretaggers (e.g., for labeling dates and title words) and zoners (e.g., for word tokenization, sentence boundary determination and headline segmentation)." [Aberdeen at 141.] "Once the initial phrasing has taken place, the phraser proceeds with phrase identification proper. This is driven by a sequence of phrase-finding rules. Each rule in the sequence is applied in turn against all of the phrases in all the sentences under analysis. If the antecedents of the rule are satisfies by a phrase, then the action indicated by the rule is executed immediately. The action can either change the label of the satisfying phrase, grow its boundaries, or create new phrases." [Aberdeen at 144.] "In support of the syntactic phrase finder, or phraser as we call it, the input text must be tagged for part-of-speech. This part-of-speech tagging is the principal role of the UNIX preprocess, and it is itself supported by a number of pretaggers (e.g., for labeling dates and title words) and zoners (e.g., for word tokenization, sentence boundary determination and headline segmentation)." [Aberdeen at 141.] "Rules can test lexemes to the left and right of the phrase, or they can look at the lexemes in the phrase. Tests in turn can be part-of-speech queries, literal lexeme matches, tests for presence of neighboring phrases, or the application of predicates that are evaluated by invoking a Lisp procedure. There are several reasons for keeping this rule language simple. In the case of hand-

U.S. PATENT NO. 6,516,321	DISCLOSURE
	crafted rules, it facilitates the process of designing a rule sequence. In the case of machine-learned rules, it restricts the size of the search space on each epoch of the learning regimen, thus making it tractable." [Aberdeen at 144.] "It is important to note that search strategy in the phraser differs significantly from that in standard parsers. In standard parsing, one searches for any and all rules whose antecedents might apply given the state of the parser's chart: all these rules become candidates for application, and indeed they all are applied (modulo higher-order search control.) In our parser, only the current rule sequence is tested: the rule is applied wherever this test succeeds, and the rule is never revisited at any subsequent stage of processing. After the final rule of a sequence is run, no further processing occurs." [Aberdeen at 144.] "To make this clearer, consider a simple name entity rule as applied to identifying persons. (def-phraser label none left-1 phrase-ttl label-action person) This rule changes the label of a phrase from none to person if the phrase is bordered on its left by a ttl phrase. On the sample sentence, this rule causes the following relabeling of the phrase around 'James'. Yesterday, <none>McCann</none>made official what had been widely anticipated: <ttl>Mr.</ttl><person>James</person>, <num>57</num>years old, is stepping down as <post>chief executive</post> on <date>July 1</date> and will retire as <post>chairman</post> at the end of the year. Once this rule has run, the labelings it instantiates become available as input to subsequent rules in the sequence, e.g., rules that attach the title to the person in 'Mr. James', that attach the age apposition, and so forth." [Aberdeen at 145.]
[ii] a separate tag pair for each of a plurality of different information types	Aberdeen discloses separate tags for each plurality of different information types. For example, the Alembic system tags several different information types, such as numbers, posts, titles, dates and names. Each information type has a separate tag pair. For example [Aberdeen at 145]: *Yesterday, <none>McCann</none>made official what had been widely anticipated: <ttl>Mr.</ttl><person>James</person>, <num>57</num>years old, is stepping down as <post>chief executive</post> on <date>July 1</date> and will retire as <post>chairman</post> at the end of the year.* "To illustrate the process, consider the following walkthrough sentence, as tagged by the NE rule sequence. *But the bragging rights to <org>Coke</org>'s ubiquitous advertising belongs to <org>Creative Artists Agency</org>, the big <location>Hollywood</location>talent agency.* The org label on "Creative Artists Agency" was set by a predicate that tests for org keywords (like "Agency"). 'Coke' was found to be an org elsewhere in the document, and the label was then percolated. Finally, the location label on 'Hollywood' was set be a predicate that inspects the tried-and-not-so-true TIPSTER gazetteer." [Aberdeen at 145.]

U.S. PATENT NO. 6,516,321	DISCLOSURE
	"The preprocess includes specialized phrase taggers. The *title-tagger* makes personal titles, making distinctions along the lines drawn by NE and ST tasks. Included are personal honorifics (Dr., Mrs.); military and religious titles (Vicar, Sgt.); corporate posts (CEO, chairman); and "profession" words (analyst, spokesperson). The *date-tagger* identifies TIMEX phrases. It uses a lex-based scanner as a front-end for tokenizing and typing its input; then pattern-matching engine finds the actual date phrases." [Aberdeen at 142.]
[iii] and a separate search rule for each pair,	Aberdeen discloses a separate search rule for each pair of subject matter specific tags, as discussed above in connection with feature [i].
[iv] each pair including a begin tag and an end tag, the method comprising the steps of:	Aberdeen discloses that each pair of subject matter specific tags has a begin tag and an end tag, as discussed above in connection with features [i] and [ii].
(a) receiving a record**;**	Aberdeen discloses that the method implemented by the Alembic program involves receiving a record. For example: "Prior to the part-of-speech tagger, however, a text to be processed by Alembic passes through several preprocess stages; each preprocessor 'enriches' the text by means of SGML tags." [Aberdeen at 142.]
(b) examining the record according to the search rules to identify record segments including information of each of the information types;	Aberdeen discloses that the method implemented by the Alembic program involves examining the received record according to search rules to identify record segments including information of each of the information types, as discussed above in connection with feature [i]. See also: "Prior to the part-of-speech tagger, however, a text to be processed by Alembic passes through several preprocess stages; each preprocessor 'enriches' the text by means of SGML tags." [Aberdeen at 142.] "The preprocess includes specialized phrase taggers. The title-tagger makes personal titles, making distinctions along the lines drawn by NE and ST tasks. Included are personal honorifics (Dr., Mrs.); military and religious titles (Vicar, Sgt.); corporate posts (CEO, chairman); and "profession" words (analyst, spokesperson). The date-tagger identifies TIMEX phrases. It uses a lex-based scanner as a front-end for tokenizing and typing its input; then pattern-matching engine finds the actual date phrases." [Aberdeen at 142.] "The Alembic phrase finder, or phraser for short, performs the bulk of the system's syntactic analysis. As noted above, it has somewhat less recognition power than a finite-state machine, and as such shares many characteristics of pattern-matching systems, such as Circus [10] or FASTUS[2]. Where it differs from these systems is in being driven by rule sequences." [Aberdeen at 144.] "The phraser process operates in several steps. First, a set of initial phrasing functions is applied to all of the sentences to be analyzed. These functions are responsible for seeding the sentences with likely candidate phrases of various kinds. This seeding process is driven by word lists, part-of-speech information, and pre-taggings provided by the processors. Initial phrasing produces a number of phrase structures, many of which have the initial null labeling (none), while some have been assigned an initial label (e.g. num)." [Aberdeen at 144.]
(c) when a record segment is identified which is of a particular information type:	Aberdeen discloses that the method implemented by the Alembic program, when it identifies a record segment of a particular information type, involves

U.S. Patent No. 6,516,321	Disclosure
accessing the tag pair associated with the information type; inserting the begin tag before the identified segment and inserting the end tag after the identified segment.	accessing the tag pair associated with the information type and inserting the begin tag before the identified segment and the end tag after the identified segment, as discussed above in connection with features [i] and [ii]. The following is an example of the tagged material created by Alembic: *Yesterday, <none>McCann</none>made official what had been widely anticipated: <ttl>Mr.</ttl><person>James</person>, <num>57</num>years old, is stepping down as <post>chief executive</post> on <date>July 1</date> and will retire as <post>chairman</post> at the end of the year.*