(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
1 December 2005 (01.12.2005)

PCT

(10) **International Publication Number**
WO 2005/114494 A1

(51) **International Patent Classification**[7]: **G06F 17/30**

(21) **International Application Number:** PCT/US2005/017979

(22) **International Filing Date:** 19 May 2005 (19.05.2005)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
60/573,513 21 May 2004 (21.05.2004) US

(71) **Applicant** *(for all designated States except US)*: **COMPUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, NY 11749 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **BETTS, Christopher** [AU/AU]; 10 Ramage Road, Mount Dandenong, Victoria 3767 (AU). **ROGERS, Tony** [AU/AU]; P.O. Box 2232, Rowville, Victoria 3178 (AU).

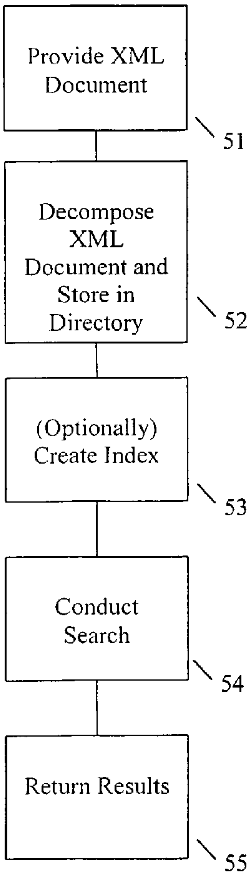
(74) **Agents: PEDERSEN, Luke, K.** et al.; Baker Botts L.L.P., 2001 Ross Avenue, Suite 600, Dallas, TX 75201-2980 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) **Title:** STORING MULTIPART XML DOCUMENTS


Provide XML Document
51
Decompose XML Document and Store in Directory
52
(Optionally) Create Index
53
Conduct Search
54
Return Results
55

(57) **Abstract:** A method for storing an XML document, the method including decomposing the XML document into a hierarchy of nodes and creating an index of the nodes.

ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

STORING MULTIPART XML DOCUMENTS

BACKGROUND

Reference to Related Application

The present disclosure is based on and claims the benefit of Provisional Application Serial No. 60/573,513 filed May 21, 2004, the entire contents of which are herein incorporated by reference.

TECHNICAL FIELD

The present disclosure relates to XML documents and, more specifically, to storing multipart XML documents.

DESCRIPTION OF THE RELATED ART

Extensible Markup Language (XML) is a popular markup language capable of describing many different types of data, by which software applications can communicate with one another and with human users. XML is a human readable markup language. XML instructions can be read and understood by a human, in contrast to, for example, computer languages that store and send information as binary data.

Because XML is a human readable, platform independent standard, data formulated in XML may be read and interpreted by any computer utilizing any platform, and is easily manipulated and tested by software developers. The ease of use of XML across many different hardware and software environments makes it a popular choice for communication between modern computer software, especially web services based applications.The growing popularity of web services based applications and XML has created a growing demand for new and useful methods to make the programming of web based applications quicker, easier and more efficient. The Apache Cocoon Project (cocoon) is one example of such a system. Cocoon is a web development framework built around the concept of separation of concerns and component-based web development. Cocoon seeks to provide a framework by which a set of components may be programmed where each component provides an isolated function. These components may then be used as building blocks for the development of more complex components and/or complete web services based applications. Using the cocoon framework, users may be able to hook together a series of pre-developed components to form a web based application without needing to engage in the minute details of computer programming. Cocoon also allows users to extract added benefit from programmed components by reusing them in a wide range of web applications without reprogramming what has previously been programmed.

XML document tools have been developed for the storing and searching of data. For example, Lightweight Directory Access Protocol (LDAP) provides for the accessing of on-line directory services. As part of the cocoon framework, a method is provided for the storing and searching of XML document data. Cocoon provides for a database implementation that allows for the manual indexing of particular elements of an XML document, allowing for the XML document to be located during a search. However, cocoon's ability to search for particular subdocuments of an XML document is very limited.

Other methods for searching through XML documents exist. For example, XML documents may be searched through linearly, one after another. However, such a non-indexed search can take a long time, especially in light of the long length of XML documents attributable to their human readable nature.

SUMMARY

A method for storing an XML document, the method including decomposing the XML document into a hierarchy of nodes, and creating an index of the nodes.

A method for searching for an XML document, the method including searching for the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

A system for storing an XML document, includes a decomposing unit for decomposing the XML document into a hierarchy of nodes, and a creating unit for creating an index of the nodes.

A system for searching an XML document, includes a searching unit for searching the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

A computer system includes a processor and a computer recording medium including computer executable code executable by the processor for storing an XML document. The computer executable code includes code for decomposing the XML document into a hierarchy of nodes, and code for creating an index of the nodes.

A computer system includes a processor and a computer recording medium including computer executable code executable by the processor for searching an XML document. The computer executable code includes code for searching the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present disclosure and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

FIG. 1 shows an example of an XML document that may be stored and searched according to an embodiment of the present disclosure;

FIG. 2 shows a diagram of how the example XML document that may be stored and searched according to an embodiment of the present disclosure is organized;

FIG. 3 shows how the example XML document may be decomposed according to an embodiment of the present disclosure;

FIG. 4 shows the text of the nodes that the example XML document may be decomposed into according to an embodiment of the present disclosure;

FIG. 5 shows a flowchart illustrating an embodiment of the present disclosure; and

FIG. 6 shows an example of a computer system capable of implementing the method and apparatus according to embodiments of the present disclosure.

DETAILED DESCRIPTION

In describing the preferred embodiments of the present disclosure illustrated in the drawings, specific terminology is employed for sake of clarity. However, the present disclosure is not intended to be limited to the specific terminology so selected, and it is to be understood that each specific element includes all technical equivalents which operate in a similar manner.

The present disclosure describes a system and method for indexing and searching through stored XML documents and XML sub documents. According to embodiments of the present disclosure, searching through stored XML documents and sub documents may be quick and may minimize processor overhead, for example, it may be computationally efficient. According to embodiments of the present disclosure, search results may be returned as the actual text of the XML document. This may offer advantages, for example, the retrieval of actual XML text may facilitate XML cryptography.

Figure 5 shows a flowchart illustrating an embodiment of the present disclosure. First, an XML document may be provided (Step S51). Figure 1 shows an example of an XML document 11 that may be provided to be stored and searched according to an embodiment of the present disclosure. The XML document 11 defines "presents" as including a dragon toy that has a green color and a fluffy type, a horse toy that has a green color and a hairy type, and chocolate food. Embodiments of the present disclosure will be described with respect to this example XML document. Of course, the same methods and systems may be applied, more generally, to any XML document or document.

Figure 2 shows a diagram of how the example XML document 11 to be stored and searched according to an embodiment of the present disclosure is organized. The complete XML document 21 defines presents. Within this document 21, three sub documents 22, 23, 24 may be found. A sub document is a portion of a document that may be hierarchically below a document or sub document and/or may be capable of independent function. The first sub document 22 defines a dragon. The dragon sub document 22 is defined as a toy having a green color 25 and a fluffy type 26. The green color 22 and fluffy type 26 are themselves sub documents that define individual characteristics. For example, the green color sub document 25 defines green as a color.

The second sub document 23 defines a horse. The horse sub document 23 is defined as a toy having a blue color 27 and a hairy type 28. The blue color 27 and the hairy type 28 are themselves sub documents that define individual characteristics.

The third sub document 24 defines chocolate. The chocolate sub document 24 is defined as a food but has no sub documents defined within it.

After an XML document has been provided (Step S51), the provided XML document may be decomposed (Step S52). Decomposing an XML document may be the process of dividing the program into discrete units that can be searched. Figure 3 shows how the example XML document 11 may be decomposed according to an embodiment of the present disclosure. According to an embodiment of the present disclosure, an XML document may be decomposed into one or more nodes. Figure 4 shows the text of the nodes that the example XML document may be decomposed into according to an embodiment of the present disclosure. These nodes may be hierarchical whereby a parent node may have one or more children nodes. The first node generally represents the entire XML document.

Along with the XML document or sub document text, each node may contain a unique directory identifier. The unique directory identifier may list a node number as well as node numbers for the parent of the node and all other ancestors, where they may exist. Each node may contain a node element name. The node element name may be, for example, a name for the node. This name, for example, may be used to identify the node in an index (directory) that may be created to facilitate searches. Each node may contain one or more node attributes. These attributes may be a list of any characteristics of the node that may facilitate searching for that particular document or sub document. The inclusion of node attributes is an optional feature of the present disclosure and other embodiments of the present disclosure do not use node attributes. Where desired, node attributes may be manually entered by a user or automatically generated by analyzing the text of the document or sub document. Additionally, other values may be stored as part of the node. For example, other values may be stored to facilitate searching.

The first node 31, according to the example XML document 11, represents the entire example XML document 11. The hierarchical position of this node 31 is shown in Figure 3. The full text 41 of the first node 31 is shown in Figure 4. In addition to containing the full text of the XML document 11, the first node 31 contains a node element name, here "Node1" and a unique directory identifier, here Element=1. Because only one element is listed as the unique directory identifier, it is clear that this node represents the top of the hierarchy, for example, the full XML document.

Complete XML documents, for example, nodes representing the top of the hierarchy, may list the full XML document as the program text as shown in Figures 3 and 4. XML sub documents, for example, nodes that may be in a position other than the top of the hierarchy, for example, children of the complete XML document or children of other sub documents, may list the XML document section that defines that sub document including all sub documents of that sub document, if any should exist, as well as all other sub documents that may be descendents of that sub document.

For example, the second node 32 represents a sub document of the example XML document 11 that defines the toy dragon. The text 42 of the second node 32 is shown in Figure 4. In addition to including the XML text that defines the dragon as a toy, the text of the second node 32 includes the XML text that defines the color green, which is its self a node 35 and an XML sub document, and the XML text that defines the type fluffy, which is its self a node 36 and an XML sub document.

In addition to including the XML text of the toy dragon sub document, the text 42 of the second node 32, contains a node element name, here "Node2" and a unique directory identifier, here Element=2, Element=1. Because two elements are listed as the unique directory identifier, it is clear that this node is the second node and is a child of the first node on the hierarchy.

Likewise the third node through the eighth node 33-38 all contain the definitions of their particular sub document along with all of the sub documents that may be their respective descendents. The text of these sub documents 43-48 respectively contains the XML text that defines their particular sub document along with all of the sub documents that may be their respective descendents along with their respective node element names and unique identifiers.

After an XML document has been decomposed into nodes representing the XML document and all sub documents as a hierarchy of nodes as described above (Step S52), each node may be stored in a hierarchical directory of XML documents and sub documents. A searchable index may be created to facilitate searching of the directory (Step S53). Each node may be indexed according to its attributes. For example, each node may be indexed according to its XML text, unique directory identifier, node element name, and/or any other attributes that may be listed within each node. For example, each node may be indexed according to its hierarchy.

The storing of the XML document and sub document nodes and the creation of an index may be carried out according to known techniques for storing information to a directory and creating a searchable index. For example, the techniques used for handling LDAP directories may be used to handle the creation of a searchable index.

The above directory may be searched for a particular XML document or sub document (Step S54). To facilitate the search, the created searchable index may be used. After the search has been performed, search results may be returned (Step S55). For example, the full XML text of the documents and/or sub documents resulting from the search may be displayed.

Figure 6 shows an example of a computer system which may implement the method and system of the present disclosure. The system and method of the present disclosure may be implemented in the form of a software application running on a computer system, for example, a mainframe, personal computer (PC), handheld computer, server, etc. The software application may be stored on a recording media locally accessible by the computer system and accessible via a hard wired or wireless connection to a network, for example, a local area network, or the Internet.

The computer system referred to generally as system 1000 may include, for example, a central processing unit (CPU) 1001, random access memory (RAM) 1004, a printer interface 1010, a display unit 1011, a local area network (LAN) data transmission controller 1005, a LAN interface 1006, a network controller 1003, an internal buss 1002, and one or more input devices 1009, for example, a keyboard, mouse etc. As shown, the system 1000 may be connected to a data storage device, for example, a hard disk, 1008 via a link 1002.

The above specific embodiments are illustrative, and many variations can be introduced on these embodiments without departing from the spirit of the disclosure or from the scope of the appended claims. For example, elements and/or features of different illustrative embodiments may be combined with each other and/or substituted for each other within the scope of this disclosure and appended claims.

What is claimed is:

1. A method for storing an XML document, the method comprising:

decomposing the XML document into a hierarchy of nodes; and

creating an index of the nodes.

2. The method of claim 1, wherein decomposing the XML document into a hierarchy of nodes comprises storing the XML document as a node at a highest level of the hierarchy of nodes.

3. The method of claim 2, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more portions of the XML document as nodes of the hierarchy of nodes, wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it in the hierarchy of nodes.

4. The method of claim 2, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

5. The method of claim 4, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

6. The method of claim 5, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

7. The method of claim 1, wherein each node of the hierarchy of nodes contains a unique directory identifier.

8. The method of claim 1, wherein each node of the hierarchy of nodes contains a node element name.

9. The method of claim 1, wherein each node of the hierarchy of nodes contains one or more node attributes.

10. A method for searching an XML document, the method comprising:

searching the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

11. The method of claim 10, wherein one or more of the one or more XML documents is a sub document.

12. The method of claim 10, wherein decomposing an XML document of the one or more XML documents into a hierarchy of nodes comprises storing the XML document as a node at a highest level of the hierarchy of nodes.

13. The method of claim 12, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more portions of the XML document as nodes of the hierarchy of nodes wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it on the hierarchy of nodes.

14. The method of claim 12, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

15. The method of claim 14, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

16. The method of claim 15, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

17. The method of claim 10, wherein each node of the hierarchy of nodes contains a unique directory identifier.

18. The method of claim 10, wherein each node of the hierarchy of nodes contains a node element name.

19. The method of claim 10, wherein each node of the hierarchy of nodes contains one or more node attributes.

20. A system for storing an XML document, comprising:

a decomposing unit for decomposing the XML document into a hierarchy of nodes; and

a creating unit for creating an index of the nodes.

21. The system of claim 20, wherein decomposing the XML document into a hierarchy of nodes comprises storing the XML document as a node at a highest level of the hierarchy of nodes.

22. The system of claim 21, wherein decomposing the XML document into a

hierarchy of nodes comprises storing one or more portions of the XML document as nodes of the hierarchy of nodes wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it on the hierarchy of nodes.

23. The system of claim 21, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

24. The system of claim 23, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

25. The system of claim 24, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

26. The system of claim 20, wherein each node of the hierarchy of nodes contains a unique directory identifier.

27. The system of claim 20, wherein each node of the hierarchy of nodes contains a node element name.

28. The system of claim 20, wherein each node of the hierarchy of nodes contains one or more node attributes.

29. A system for searching an XML document, comprising:

a searching unit for searching the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

30. The system of claim 29, wherein one or more of the one or more XML documents is a sub document.

31. The system of claim 29, wherein decomposing an XML document of the one or more XML documents into a hierarchy of nodes comprises storing the XML document as a node at a highest level of the hierarchy of nodes.

32. The system of claim 31, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more portions of the XML document as nodes of the hierarchy of nodes, wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it on the hierarchy of nodes.

33. The system of claim 31, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

34. The system of claim 33, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

35. The system of claim 34, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

36. The system of claim 29, wherein each node of the hierarchy of nodes contains a unique directory identifier.

37. The system of claim 29, wherein each node of the hierarchy of nodes contains a node element name.

38. The system of claim 29, wherein each node of the hierarchy of nodes contains one or more node attributes.

39. A computer system comprising:

a processor; and

a computer recording medium including computer executable code executable by the processor for storing an XML document, the computer executable code comprising:

code for decomposing the XML document into a hierarchy of nodes; and

code for creating an index of the nodes.

40. The computer system of claim 39, wherein the code for decomposing the XML document into a hierarchy of nodes comprises code for storing the XML document as a node at a highest level of the hierarchy of nodes.

41. The computer system of claim 40, wherein the code for decomposing the XML document into a hierarchy of nodes comprises code for storing one or more portions of the XML document as nodes of the hierarchy of nodes wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it on the hierarchy of nodes.

42. The computer system of claim 40, wherein the code for decomposing the XML document into a hierarchy of nodes comprises code for storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

43. The computer system of claim 42, wherein the code for decomposing the XML document into a hierarchy of nodes comprises code for storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

44. The computer system of claim 43, wherein the code for decomposing the XML document into a hierarchy of nodes comprises code for storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

45. The computer system of claim 39, wherein each node of the hierarchy of nodes contains a unique directory identifier.

46. The computer system of claim 39, wherein each node of the hierarchy of nodes contains a node element name.

47. The computer system of claim 39, wherein each node of the hierarchy of nodes contains one or more node attributes.

48. A computer system comprising:

a processor; and

a computer recording medium including computer executable code executable by the processor for searching an XML document, the computer executable code comprising:

code for searching for the XML document using an index that has been created by decomposing one or more XML documents into a hierarchy of nodes.

49. The computer system of claim 48, wherein one or more of the one or more XML documents is a sub document.

50. The computer system of claim 48, wherein decomposing an XML document of the one or more XML documents into a hierarchy of nodes comprises storing the XML document as a node at a highest level of the hierarchy of nodes.

51. The computer system of claim 50, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more portions of the XML document as nodes of the hierarchy of nodes wherein each of the nodes of the hierarchy of nodes that is not the highest level of the hierarchy of nodes is a portion of the node immediately above it on the hierarchy of nodes.

52. The computer system of claim 50, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more first portions of the XML document as nodes at a second-highest level of the hierarchy of nodes.

53. The computer system of claim 52, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more second portions of the XML document which are also a portion of the corresponding first portion of the XML document, as nodes at a third-highest level of the hierarchy of nodes.

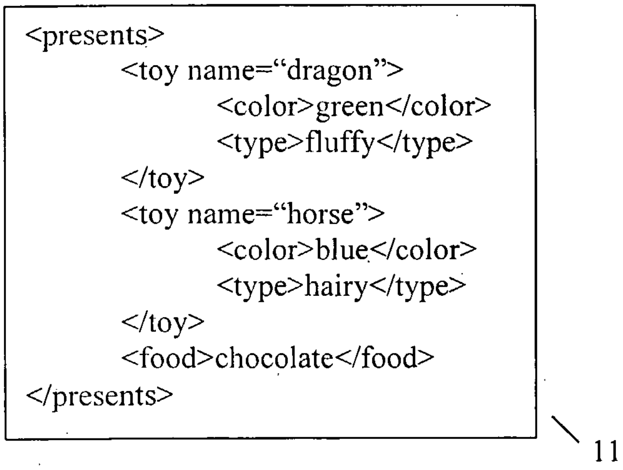
54. The computer system of claim 53, wherein decomposing the XML document into a hierarchy of nodes comprises storing one or more subsequent portions of the XML document which are also portions of the corresponding portion of the XML document directly above the corresponding one or more subsequent portions of the XML document in the hierarchy of nodes, as a node at a level of the hierarchy immediately below the corresponding one or more subsequent portions of the XML document.

55. The computer system of claim 48, wherein each node of the hierarchy of nodes contains a unique directory identifier.

56. The computer system of claim 48, wherein each node of the hierarchy of nodes contains a node element name.

57. The computer system of claim 48, wherein each node of the hierarchy of nodes contains one or more node attributes.

Fig. 1



```
<presents>
        <toy name="dragon">
                <color>green</color>
                <type>fluffy</type>
        </toy>
        <toy name="horse">
                <color>blue</color>
                <type>hairy</type>
        </toy>
        <food>chocolate</food>
</presents>
```

Fig. 2

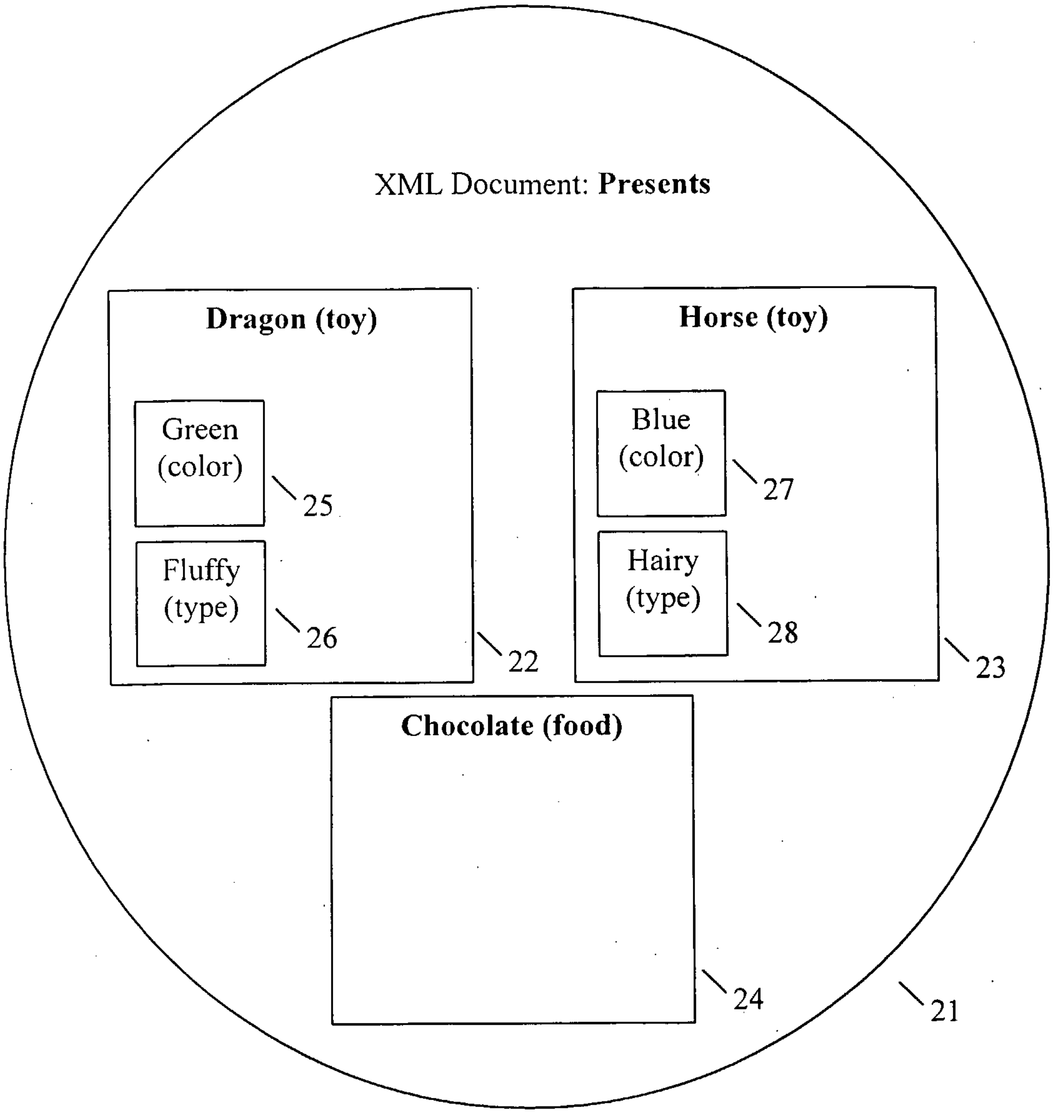

XML Document: **Presents**
Dragon (toy)
Green (color)
25
Fluffy (type)
26
22
Horse (toy)
Blue (color)
27
Hairy (type)
28
23
Chocolate (food)
24
21

Fig. 3

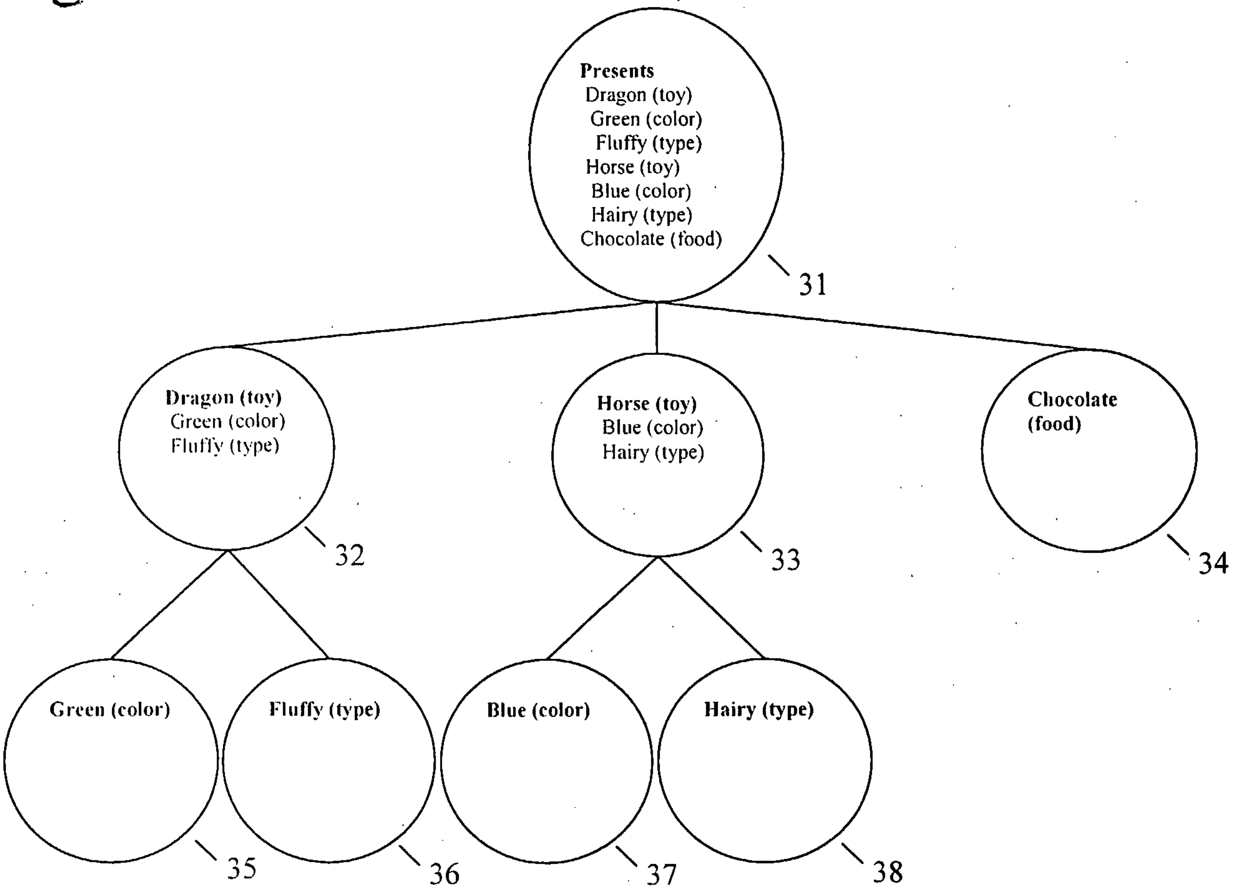
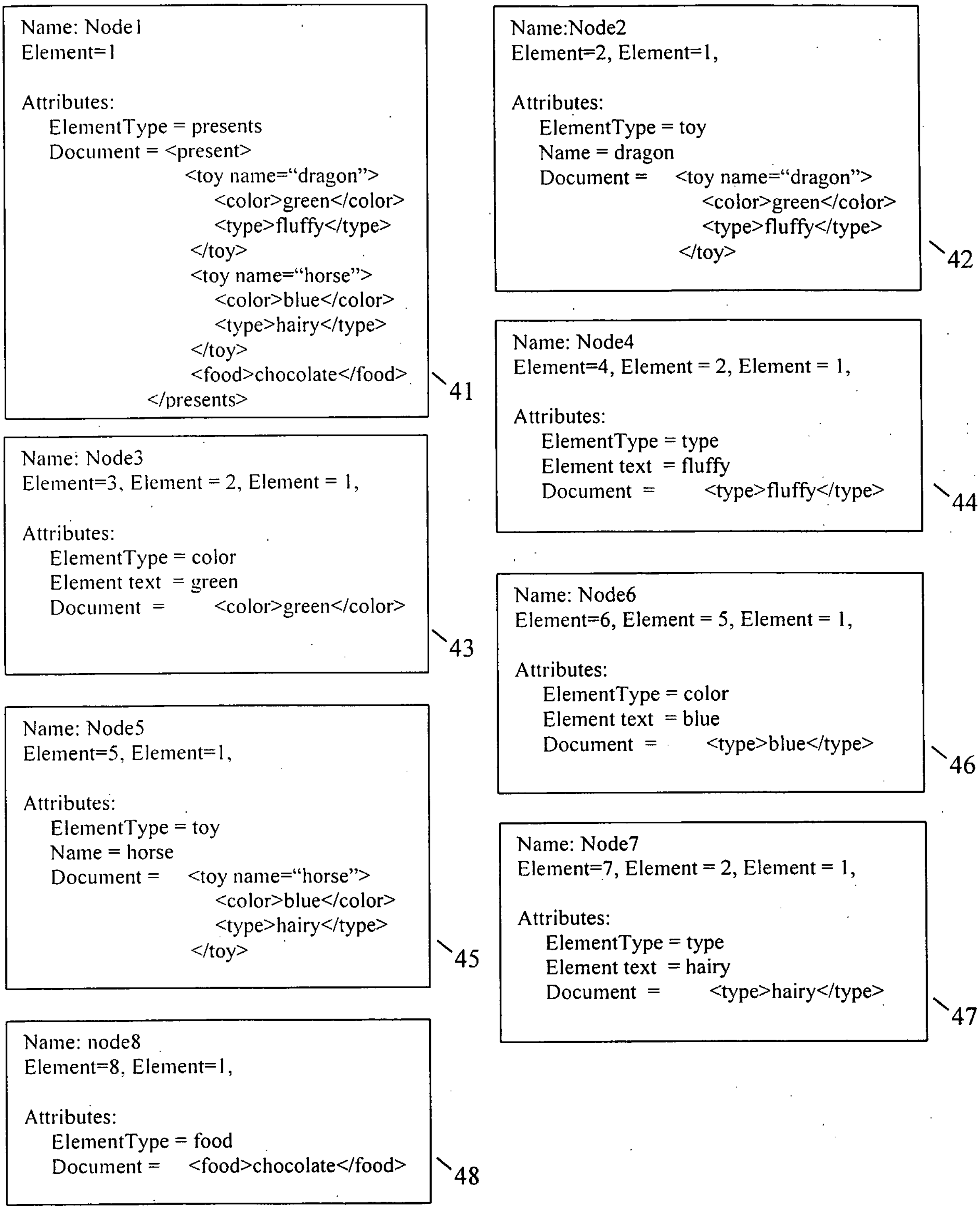

Presents
Dragon (toy)
Green (color)
Fluffy (type)
Horse (toy)
Blue (color)
Hairy (type)
Chocolate (food)
31
Dragon (toy)
Green (color)
Fluffy (type)
32
Horse (toy)
Blue (color)
Hairy (type)
33
Chocolate
(food)
34
Green (color)
35
Fluffy (type)
36
Blue (color)
37
Hairy (type)
38

Fig. 4



```
Name: Node1
Element=1

Attributes:
   ElementType = presents
   Document = <present>
                  <toy name="dragon">
                     <color>green</color>
                     <type>fluffy</type>
                  </toy>
                  <toy name="horse">
                     <color>blue</color>
                     <type>hairy</type>
                  </toy>
                  <food>chocolate</food>
              </presents>
```

```
Name: Node3
Element=3, Element = 2, Element = 1,

Attributes:
   ElementType = color
   Element text  = green
   Document  =      <color>green</color>
```

```
Name: Node5
Element=5, Element=1,

Attributes:
   ElementType = toy
   Name = horse
   Document =    <toy name="horse">
                    <color>blue</color>
                    <type>hairy</type>
                 </toy>
```

```
Name: node8
Element=8, Element=1,

Attributes:
   ElementType = food
   Document =    <food>chocolate</food>
```

```
Name:Node2
Element=2, Element=1,

Attributes:
   ElementType = toy
   Name = dragon
   Document =    <toy name="dragon">
                    <color>green</color>
                    <type>fluffy</type>
                 </toy>
```

```
Name: Node4
Element=4, Element = 2, Element = 1,

Attributes:
   ElementType = type
   Element text  = fluffy
   Document  =      <type>fluffy</type>
```

```
Name: Node6
Element=6, Element = 5, Element = 1,

Attributes:
   ElementType = color
   Element text  = blue
   Document  =      <type>blue</type>
```

```
Name: Node7
Element=7, Element = 2, Element = 1,

Attributes:
   ElementType = type
   Element text  = hairy
   Document  =      <type>hairy</type>
```

Fig. 5

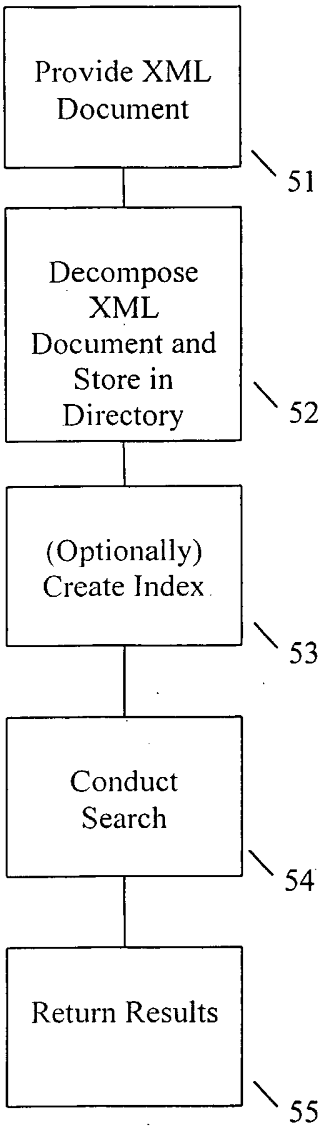
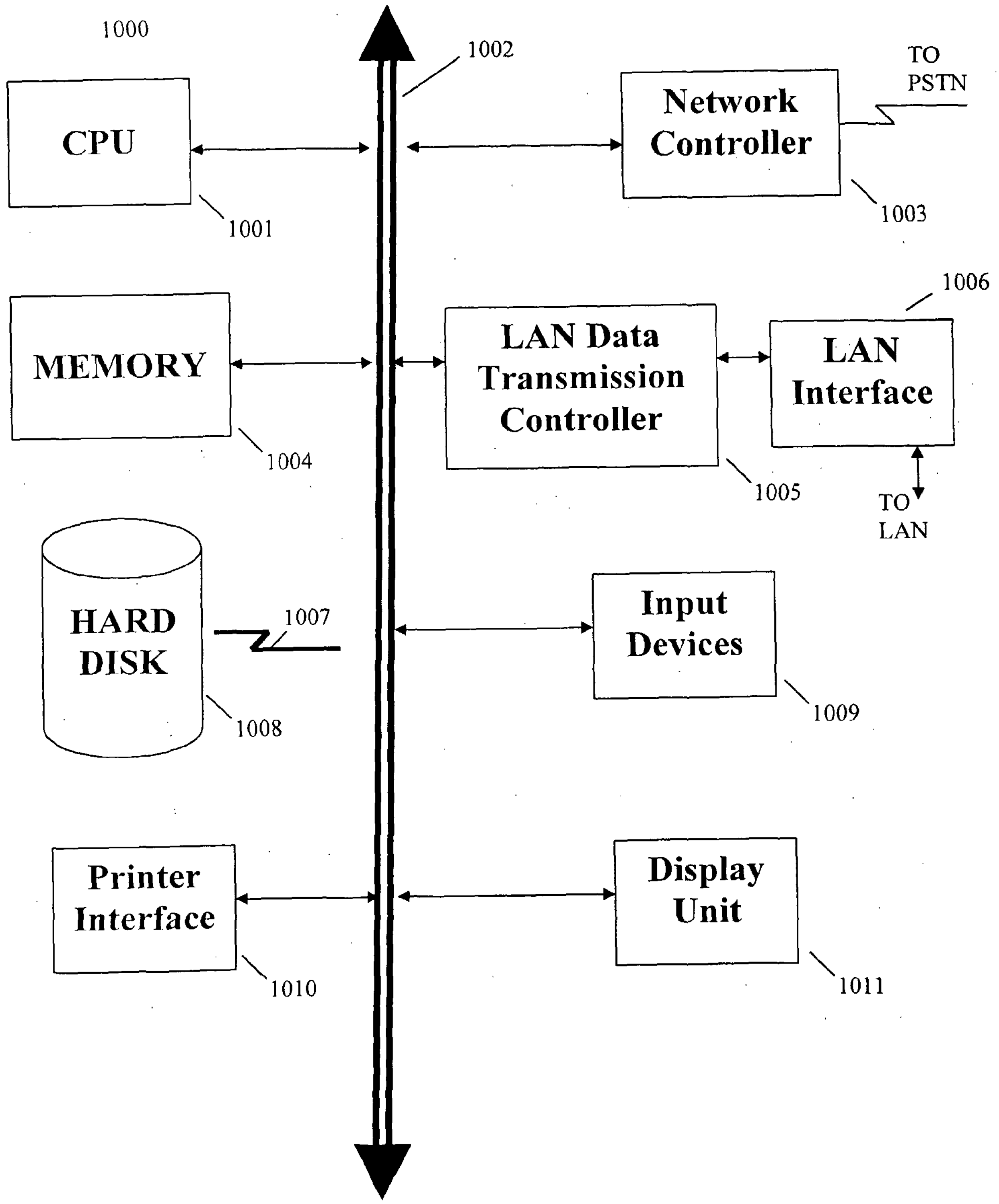

Provide XML Document
51
Decompose XML Document and Store in Directory
52
(Optionally) Create Index
53
Conduct Search
54
Return Results
55

Fig. 6


1000
CPU
1001
1002
Network Controller
1003
TO PSTN
MEMORY
1004
LAN Data Transmission Controller
1005
LAN Interface
1006
TO LAN
HARD DISK
1007
1008
Input Devices
1009
Printer Interface
1010
Display Unit
1011

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/017979

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00/45304 A (OBJECT DESIGN, INC) 3 August 2000 (2000-08-03) abstract; figures 2-8 page 7, line 3 - line 24 page 9 - page 19	1-57
X	US 2003/101169 A1 (BHATT VADIRAJA ET AL) 29 May 2003 (2003-05-29) abstract; figures 3-8 paragraph ‘0033! - paragraph ‘0140!	1-57
X	US 2004/073541 A1 (LINDBLAD CHRISTOPHER ET AL) 15 April 2004 (2004-04-15) the whole document	1-57
X	US 2002/143742 A1 (NONOMURA KATSUHIKO ET AL) 3 October 2002 (2002-10-03) the whole document	1-57
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 18 October 2005	Date of mailing of the international search report 25/10/2005
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Nazzaro, A

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/204515 A1 (SHADMON MOSHE ET AL) 30 October 2003 (2003-10-30) abstract; figures 1-14 paragraph ‘0042! - paragraph ‘0110! -----	1-57
X	US 6 681 223 B1 (SUNDARESAN NEELAKANTAN) 20 January 2004 (2004-01-20) abstract; figures 2,3 column 6 - column 17 -----	1-57
X	US 2003/033285 A1 (JALALI NEEMA ET AL) 13 February 2003 (2003-02-13) the whole document -----	1-57
X	EP 1 376 408 A (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) 2 January 2004 (2004-01-02) the whole document -----	1-57

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US2005/017979

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
WO 0045304	A	03-08-2000	AU	2633800	A	18-08-2000
			EP	1192561	A1	03-04-2002
			JP	2002536723	T	29-10-2002
US 2003101169	A1	29-05-2003	NONE			
US 2004073541	A1	15-04-2004	NONE			
US 2002143742	A1	03-10-2002	JP	2002297602	A	11-10-2002
US 2003204515	A1	30-10-2003	NONE			
US 6681223	B1	20-01-2004	NONE			
US 2003033285	A1	13-02-2003	NONE			
EP 1376408	A	02-01-2004	CN	1469276	A	21-01-2004
			US	2004044963	A1	04-03-2004