(54) Title: METHOD AND SYSTEM FOR XML MULTI-TRANSFORM
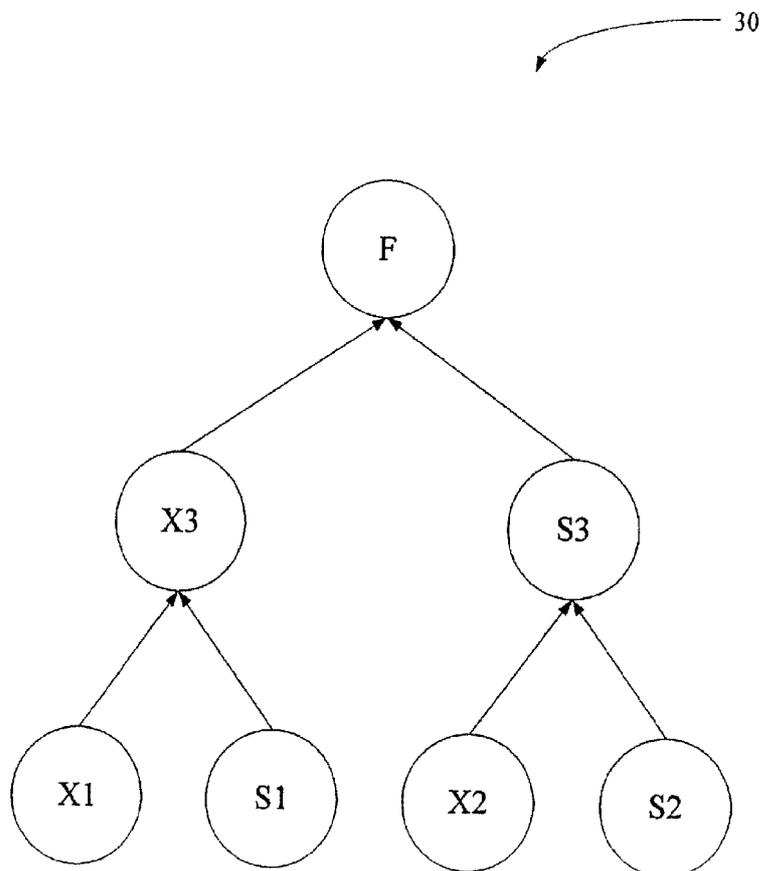
(57) Abstract: A method of performing XSLT processing is disclosed herein disclosed. The method comprises defining a binary tree, each node representing either an XML document or XSLT style sheet, each leaf representing a an existing file, and each non-leaf node having two child nodes, one representing an XML document, and one representing an XSLT style sheet, performing a post-order traversal of the binary tree, each non-leaf node being constructed by transforming the non-leaf node's respective XML document child node using the non-leaf node's respective XSLT style sheet to create the non-leaf node, and creating a final text document as the root node.

WO 2008/039929 A1

METHOD AND SYSTEM FOR XML MULTI-TRANSFORM

REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application

5   Number 60/827,111, filed September 27, 2006, and entitled "XML Multi-Transform."

BACKGROUND

[0002] The extensible markup language (XML) is a general purpose markup

language. Extensible stylesheet language (XSL) transformation (XSLT) is a language that

10   translates one XML document into another XML document, allowing for chained

translations. XSLT is itself an XML document and may be generated from another XSLT

transformation. Current extensible stylesheet language (XSL) transformation (XSLT)

processing is limited to one extensible markup language (XML) document processed by

one XSLT document. In general, custom applications are required in order to perform

15   multiple transformations of XML when multiple translations need to be performed. These

custom applications tend to be relatively complex, relative expensive, and have long

development cycles.

[0003] Thus, there is a need for a method and system of XSLT processing that is

not limited to a single XML document processed by a single XSLT document.

20

SUMMARY

[0004] Before the present methods are described, it is to be understood that this

invention is not limited to the particular methodologies or protocols described, as these

may vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the present disclosure, which will be limited only by the appended claims.

[0005] It must be noted that as used herein and in the appended claims, the

5   singular forms "a," "an," and "the" include plural reference unless the context clearly dictates otherwise. Thus, for example, reference to a "transaction" is a reference to one or more transactions and equivalents thereof known to those skilled in the art, and so forth. Unless defined otherwise, all technical and scientific terms used herein have the same meanings as commonly understood by one of ordinary skill in the art. Although any methods and

10   materials similar or equivalent to those described herein can be used in the practice or testing of the present invention, the preferred methods, devices, and materials are now described. All publications mentioned herein are incorporated herein by reference. Nothing herein is to be construed as an admission that the invention is not entitled to antedate such disclosure by virtue of prior invention.

15   [0006] In an embodiment, a method of performing XSLT processing is hereinafter disclosed. The method comprises defining a binary tree, each node representing either an XML document or XSLT stylesheet, each leaf representing a an existing file, and each non-leaf node having two child nodes, one representing an XML document, and one representing an XSLT stylesheet, performing a post-order traversal of the binary tree, each non-leaf node

20   being constructed by transforming the non-leaf node's respective XML document child node using the non-leaf node's respective XSLT stylesheet to create the non-leaf node, and creating a final text document as the root node.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Aspects, features, benefits and advantages of the present invention will be apparent with regard to the following description and accompanying drawings, of which:

[0008] **Figure 1** depicts an exemplary data structure according to an embodiment

[0009] **Figure 2** depicts an exemplary binary tree corresponding to the data structure of **Figure 1** according to a preferred embodiment.

[0010] **Figure 3** depicts a flow diagram for an exemplary method of processing XML utilizing XSLT according to an embodiment.

DETAILED DESCRIPTION

[0011] **Figure 1** illustrates various embodiments of a data structure **10** for representing multiple nested XSL transforms as a binary tree. The binary tree may be represented in an XML format with a schema comprising a root tag **12,** and four additional tags under the root tag **12.** The four tags comprise a transform tag 14, an XML tag 16, and XSL tag **18,** and a parameter tag **20.** The transform tag 14 may hold one XML tag 16, one XSL tag **18,** and any number of parameter tags **20.** Each XML tag 14 may specify either a file (i.e., a leaf node) or may have a transform tag 14 underneath it. Similarly, each XSL tag 16 may specify either a file (i.e., a leaf node) or may have a transform tag 14 underneath it. The transform tags 14 underneath the XML tag 16 may comprise XML tags 16 and/or XSL tags **18.**

[0012] The data structure **10** may represent any number of chained or piped XSLT transformations in a declarative fashion as a binary tree comprising interconnected nodes. In a preferred embodiment, for a given parent node, one child

3

may be an XML document, one child may be an XSLT sheet, and the parent node may

be a resultant XML document. Or XSLT sheet The parent node can then itself be an

XML child node or an XSLT child node of another parent node. Nodes without

children may be considered leaf nodes. The only node without a parent is the top node,

5   and may be considered the root node of the binary tree. Each node may have one or two

children. Before the commencement of the method disclosed hereinafter, only the leaf

nodes are guaranteed to exist at the beginning of the disclosed method. The

intermediate files may be created on-the-fly, as hereinafter disclosed. The data structure

10 makes processing large numbers of XML documents more convenient. A simplified

10   example of a binary tree 30 corresponding to the data structure **10** of **Figure 1 is**

illustrated in **Figure 2.** In the binary tree 30, each "X" represents an XML document,

each "S" represents an XSLT stylesheet, and "F" represents the final text document.

   [0013] Figure 3 illustrates a method of XML processing using XSLT according to a

preferred embodiment. In general, the method may be used for any application in which

15   XML processing with XSLT may be useful. For example, the method may be used for

report generation, data manipulation, graphics, web page generation, etc. First, a binary

tree comprising one ore more XML documents and multiple nest XSL transforms is defined

110. According to an embodiment, the binary tree may be defined using a structure similar

to the data structure of **Figure 1.** Then, XML document and XSL sheet transformations are

20   performed **120** with a post-order tree traversal to generate a final text document at the root

node of the binary node. In the post-order tree traversal, one of the chained or piped XSLT

transforms operates on an XML document to generate a different XML document or XSLT

transform, and so on throughout the post-order tree traversal process. Any number of

transformations may be performed to generate the final text document.

[0014] Using the binary tree 30 as an example, the post-order tree traversal proceeds as follows. First, the left child "Xl" is operated on by the right child "Sl" to generate the parent "X3." Then, the left child "X2" is operated on by the right child "S2" to generate the parent "S3." Then, "X3" is operated on by "S3" to produce the final text document, "F."

[0015] The above method may be implemented in, for example, hardware or firmware. The method may be implemented as a software application using any programming language and may be embodied permanently or temporarily in any type of machine, component, physical or virtual equipment, storage medium, or propagated signal capable of delivering instructions. The software code may be stored on a computer readable medium such as magnetic or optical memory devices.

[0016] It will be appreciated that various of the above-disclosed and other features and functions, or alternatives thereof, may be desirably combined into many other different systems or applications. It will also be appreciated that various presently unforeseen or unanticipated alternatives, modifications, variations or improvements therein may be subsequently made by those skilled in the art which are also intended to be encompassed by the disclosed embodiments.

CLAIMS

1.   A method for performing XSLT processing, the method comprising:

defining a binary tree, each node representing either an XML document or XSLT stylesheet, each leaf representing a an existing file, and each non-leaf node having two child nodes, one representing an XML document, and one representing an XSLT stylesheet;

performing a post-order traversal of the binary tree, each non-leaf node being constructed by transforming the non-leaf node's respective XML document child node using the non-leaf node's respective XSLT stylesheet to create the non-leaf node; and

creating a final text document as the root node.

2.   The method of claim 1, wherein the binary tree is represented as an XML document.

3.   The method of claim 1, wherein the binary tree contains at least two nodes representing XML documents and two nodes representing XSLT documents.

4.   The method of claim 1, wherein the final text document is a report.

5.   The method of claim 1, wherein the final text document is a web page.

6.   A computer-readable storage medium comprising instructions for causing a programmable processor to:

define a binary tree, each node representing either an XML document or XSLT

stylesheet, each leaf representing a an existing file, and each non-leaf node having two

child nodes, one representing an XML document, and one representing an XSLT

stylesheet;

5   perform a post-order traversal of the binary tree, each non-leaf node being

constructed by transforming the non-leaf node's respective XML document child node

using the non-leaf node's respective XSLT stylesheet to create the non-leaf node; and

create a final text document as the root node.

10   7.   The computer-readable storage medium of claim 6, wherein the binary tree

is represented as an XML document.

8.   The computer-readable storage medium of claim 6, wherein the binary tree

contains at least two nodes representing XML documents and two nodes representing

15   XSLT documents.

9.   The computer-readable storage medium of claim 6, wherein the final text

document is a report.

20   10.   The computer-readable storage medium of claim 6, wherein the final text

document is a web page.

10

```
<root id="F">                    ← ———— 12
      <xml id="X3">              ← ———— 14
            <xml id="X1" />      ← ———— 16
                  <parameter>
            <xslt id="S1" />     ← ———— 18
                  <parameter>    ← ———— 20
      </xml>
      <xslt id="S3">
            <xml id="X2" />
                  <parameter>
            <xslt id="S2" />
      </xslt>
</root>
```

Figure 1

30

Figure 2

Define Binary Tree
Comprising One Or
More XML Documents
And Multiple Nested
XSL Transforms
— 52

Perform
Post-Order
Tree Traversal
— 54
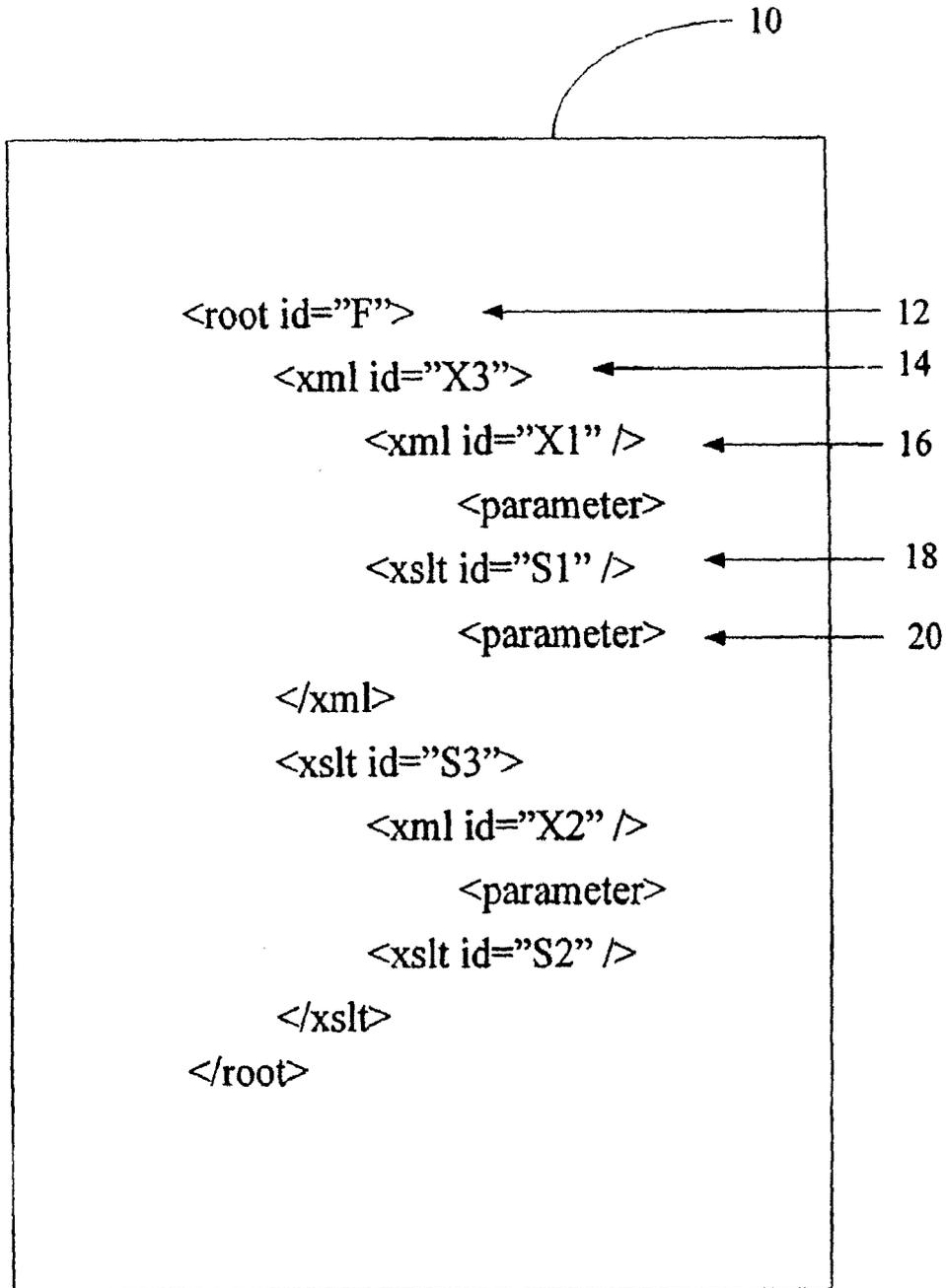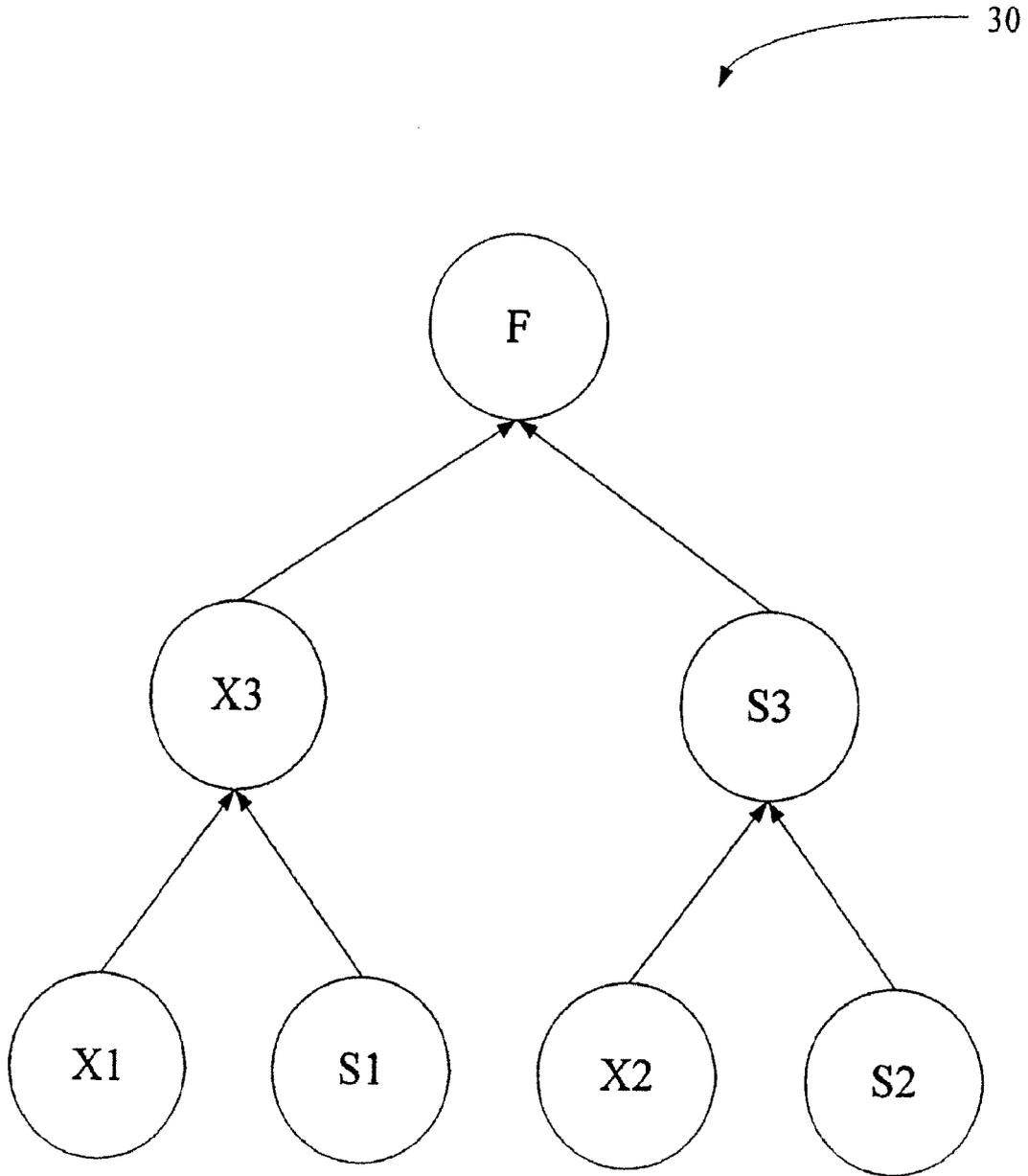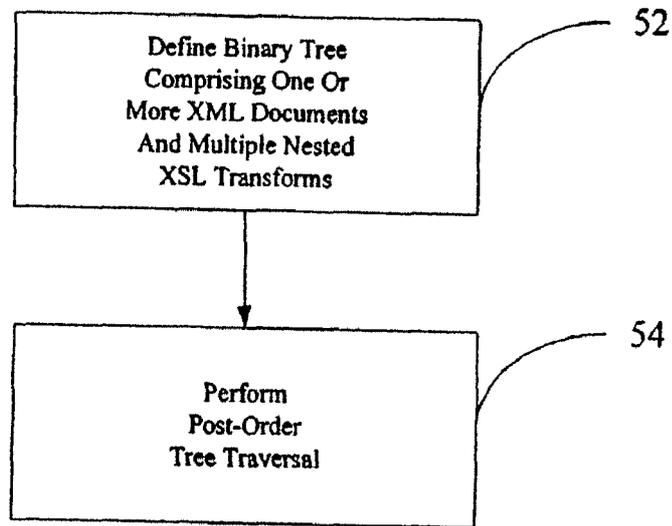
Figure 3

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
INV. 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)
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	WALSH N. ET AL: "XML Pipeline Definition Language Version 1.0" W3C NOTE, [Online] 28 February 2002 (2002-02-28), pages 1-20, XP002460302 Retrieved from the Internet: URL:http://www.w3.org/TR/xml-pipeline/> [retrieved on 2007-11-27] the whole document -----	1-10
X	WO 2004/017230 A (DIMON HUGBUNADARHUS EHF [IS]; LUDVIKSSON GEORG [IS]; HAFSTEINSSON GUDM) 26 February 2004 (2004-02-26) abstract page 22 - page 33 -----	1-10
	-/--	

[X] Further documents are listed in the continuation of Box C [X] See patent family 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 pnority 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 docu¬ ments, 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

28 November 2007

Date of mailing of the international search report

13/12/2007

Name and mailing address of the ISA/
European Patent Office, P B 5818 Patentlaan 2
NL - 228Q HV Rijswijk
Tel (+31-70) 340-2040, Tx 31 651 epo nl
Fax (+31-70) 340-3016

Authorized officer

Bernardi , Luca

4

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 2004/268238 A1 (LIU PEIYA [US] ET AL) 30 December 2004 (2004-12-30) abstract paragraph [0023] - paragraph [0071] -----	1-10

4

INTERNATIONAL SEARCH REPORT

Information on patent family members

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 2004017230	A	26-02-2004	AU 2003249590	A1	03-03-2004
			EP 1540528	A1	15-06-2005
			JP 2006510955	T	30-03-2006
US 2004268238	A1	30-12-2004	NONE		