(12) UK Patent Application (19) GB (11) 2 451 479 (13) A

(43) Date of A Publication 04.02.2009

(21) Application No: 0714906.5

(22) Date of Filing: 31.07.2007

(71) Applicant(s):
Eldon Technology Limited
(Incorporated in the United Kingdom)
Beckside Design Centre,
Millennium Business Park, Steeton,
KEIGHLEY, West Yorkshire, BD20 6QW,
United Kingdom

(72) Inventor(s):
David William Walton

(74) Agent and/or Address for Service:
Beck Greener
Fulwood House, 12 Fulwood Place,
LONDON, WC1V 6HR, United Kingdom

(51) INT CL:
H03M 7/30 (2006.01) ***G06F 17/30*** (2006.01)

(56) Documents Cited:
US 6635088 B1
Computer Networks 33 (2000) 747-765

(58) Field of Search:
INT CL **H03M**
Other: **Online: EPODOC, FULLTEXT, INSPEC, NPL, WPI, XPESP**

(54) Abstract Title: **Compressing markup language (e.g. XML) schema**

(57) A data file based on a markup language schema such as XML is encoded to reduce the file size. In the markup language schema, the information and metadata is arranged in a hierarchical structure comprised of at least one data group, to elements within the data group, and sub-elements within the elements. A text string representation of the markup language schema is produced which lists the data group, its elements and sub-elements in order. The bracketed descriptive text strings provided in the schema as both opening (e.g. <Datagroup 1>, figure 1) and closing tags are replaced by alphanumeric, alphabetic or numeric opening tags (e.g. MD01, figure 2a). Furthermore, the opening tags are used to additionally imply the close of the preceding data group, element or sub-element such that closing tags are not generally provided. By these measures compression is achieved and the size of the data file is reduced. Also included are embodiments concerning processes for transmitting and receiving data.

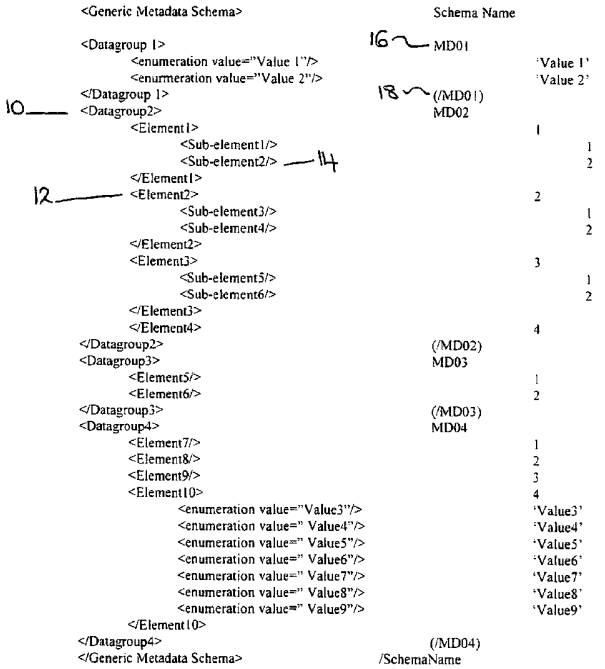

Figure 1

Figure 2a

GB 2 451 479 A

```
<Generic Metadata Schema>

<Datagroup 1>
        <enumeration value="Value 1"/>
        <enumeration value="Value 2"/>
</Datagroup 1>
<Datagroup2>
        <Element1>
                <Sub-element1/>
                <Sub-element2/>
        </Element1>
        <Element2>
                <Sub-element3/>
                <Sub-element4/>
        </Element2>
        <Element3>
                <Sub-element5/>
                <Sub-element6/>
        </Element3>
        </Element4>
</Datagroup2>
<Datagroup3>
        <Element5/>
        <Element6/>
</Datagroup3>
<Datagroup4>
        <Element7/>
        <Element8/>
        <Element9/>
        <Element10>
                <enumeration value="Value3"/>
                <enumeration value=" Value4"/>
                <enumeration value=" Value5"/>
                <enumeration value=" Value6"/>
                <enumeration value=" Value7"/>
                <enumeration value=" Value8"/>
                <enumeration value=" Value9"/>
        </Element10>
</Datagroup4>
</Generic Metadata Schema>
```

Figure 1

```
Schema Name
MD01
                'Value 1'
                'Value 2'
(/MD01)
MD02
                1
                        1       sub-element1_data
                        2       sub-element2_data
                2
                        1       sub-element3_data
                        2       sub-element4_data
                3
                        1       sub-element5_data
                        2       sub-element6_data
                4
(/MD02)
MD03
                1
                2
(/MD03)
MD04
                1               element7_data
                2               element8_data
                3               element9_data
                4
                'Value3'
                'Value4'
                'Value5'
                'Value6'
                'Value7'
                'Value8'
                'Value9'
(/MD04)
/SchemaName
```

Figure 2a

Figure 2b

```
SchemaName MD01 Value 1 MD02 1 1sub-element1_data 2 sub-element2_data 2 1 sub-element3_data 2 sub-element4_data 3 1 sub-element5_data 2 sub-element6_data MD04 1 element7_data 2 element8_data 3 element9_data /SchemaName
```

Figure 3

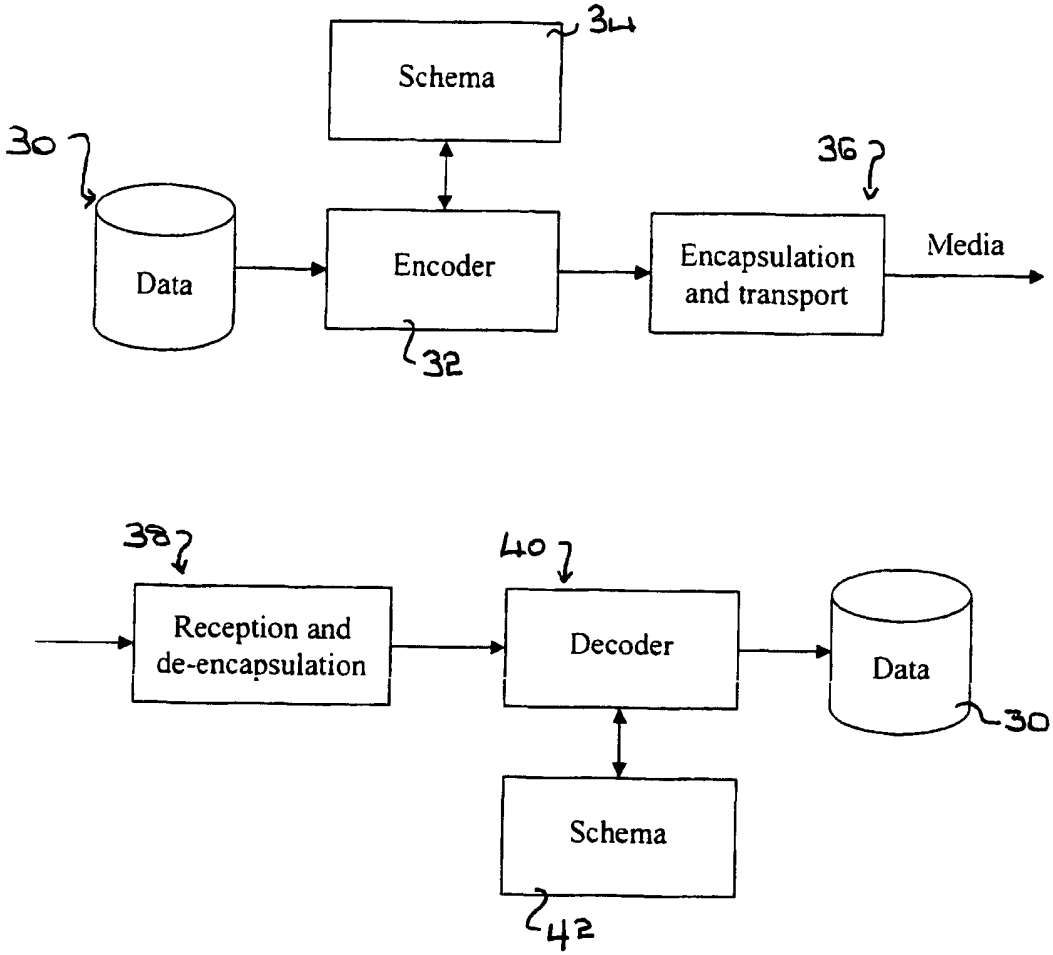
Schema
34
30
36
Data
Encoder
Encapsulation and transport
Media
32
38
40
Reception and de-encapsulation
Decoder
Data
30
Schema
42

Figure 4

IMPROVEMENTS IN OR RELATING TO A MARKUP LANGUAGE SCHEMA

The present invention relates to a text string representation of a markup language schema, to a method of encoding a data file based on a markup language schema to reduce the file size, to apparatus for encoding a data file based on a markup language schema to reduce the file size, and to processes for transmitting and receiving data.

XML, Extensible Markup Language, is a specification developed by the W3C. It is a pared down version of SGML, Standard Generalised Markup Language, designed especially for web documents. It provides a hierarchical structure for organising and tagging a document to enable the definition, transmission, validation, and interpretation of data between applications.

XML is generally considered to be the appropriate language for presenting information and metadata to be shared across networks, but it has the disadvantage that it is verbose in nature.

Schemes have been proposed for compressing XML. There have also been proposals to map XML tags to fixed values in a table in order to reduce the size of text strings.

The present invention seeks to reduce the size of a data file based on a markup language schema.

According to a first aspect of the present invention there is provided a text string representation of a markup language schema which defines a structure for carrying information and metadata, which information and metadata is identified within the schema by appropriate tags, wherein, in the text string representation, elements of information and metadata are not identified by bracketed descriptive text strings provided as both opening and closing tags but by alphanumeric, alphabetical or numeric opening tags.

As the usual bracketed descriptive text strings have been replaced by alphanumeric, alphabetical or numeric opening tags, which are smaller in size

than the bracketed descriptive text strings, the size of the data file comprised of the text string representation of the schema is reduced.

In the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements. In a preferred embodiment, the text string representation lists the data group and its elements and sub-elements in order, and the alphanumeric, alphabetical or numerical opening tag identifying each data group, element or sub-element additionally implies the close of the preceding data group, element or sub-element.

Usual markup language schema, for example the XML structure, has bracketed descriptive text strings provided in pairs to act as both opening tags and closing tags. Where an opening tag of one element is, as in this embodiment of the invention, used to also imply the end of the preceding element, there is a reduction in file size.

It has been established that it is only necessary to provide closing tags to identify the encoded data in an entire schema.

In the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements. In one embodiment, the text string representation lists the data group and its elements and sub-elements in order, and preferably alphanumeric tags are used to identify data groups and numeric tags are used to identify elements and sub-elements.

For example, alphanumeric tags identifying data groups may comprise initials and a sequentially increasing number.

Additionally and/or alternatively, sequentially increasing unique numbers may be used to identify individual elements and sub-elements.

In a preferred embodiment, a space separator is provided between an opening tag and the identified data.

Preferably, the number of available states of each alphanumeric, alphabetical or numeric opening tag must exceed the number of states to be used.

The present invention also extends to a method of encoding a data file based on a markup language schema to reduce the file size, where the schema defines a structure for carrying information and metadata which is identified within the schema by appropriate tags, where the elements of information and metadata in the schema are identified by bracketed descriptive text strings provided in pairs as opening and closing tags, the method comprising omitting the closing tags.

As closing tags are omitted, the file size is reduced.

In an embodiment, the only closing tag retained is a closing tag identifying the end of a schema.

The method preferably comprises replacing the bracketed descriptive opening tags with alphanumeric, alphabetical or numeric opening tags.

In the markup language schema the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements. In an embodiment, the method comprises listing the data group and its elements and sub-elements in order, and replacing the bracketed descriptive opening tags identifying data groups with alphanumeric tags, and replacing the bracketed descriptive opening tags identifying elements and sub-elements with numeric tags.

Preferably, the method comprises incorporating a space separator between an opening tag and the identified data.

Preferably, the encoded data file is formed into a string for transmission. The formed string may be compressed for transmission by any appropriate compression scheme.

In an embodiment, the encoding is undertaken by a client parser and a servant encoder

The present invention also extends to apparatus for encoding a data file based on a markup language schema to reduce the file size, where the schema defines a structure for carrying information and metadata which is identified within the structure by appropriate tags, where the elements of information and metadata in the schema are identified by bracketed descriptive text strings provided in pairs as opening and closing tags, the apparatus comprising a client parser to identify the defined schema and the opening and closing tags, and a server encoder to replace the identified opening and closing tags with alphanumeric, alphabetical, or numeric opening tags.

According to a further aspect of the present invention, there is provided a process for transmitting data, the process comprising encoding the data to form a data file based on a markup language schema, encapsulating the data file for transport, and transmitting the encapsulated data file, wherein the process further comprises reducing the file size of the data file before it is encapsulated using a method as defined above.

The invention also extends to a process for receiving data, the process comprising receiving and de-encapsulating an encapsulated and encoded data file to retrieve the encoded data file, and decoding the data file to obtain the data, wherein the data file had been encoded based on a markup language schema and reduced in size using a method as defined above, and wherein the data is decoded based on the markup language schema.

Embodiments of the present invention will hereinafter be described, by way of example, by reference to the accompanying drawings, in which;

Figure 1 shows the structure of a general XML metadata schema,

Figures 2a and 2b together illustrate a partially populated version of a text string representation of the markup language schema showing the reduced size of the data file,

Figure 3 shows the encoded message string developed from the text string representation of Figures 2a and 2b, and,

Figure 4 illustrates processes for transmitting and receiving data.

The invention is described further by reference to an XML metadata schema. However, the method described herein is not specific to XML language and the reduced size data files produced by the invention may be formed for any markup language having a hierarchical structure and employing opening and closing tags. Such markup languages will generally be based on the W3C rules.

Figure 1 shows the hierarchical structure of the XML schema for carrying information and metadata. Thus, the hierarchical structure comprises at least one data group 10 within which there are one or more elements 12. The elements 12 may contain one or more sub-elements 14. It will be seen that each data group 10, each element 12, and each sub-element 14 is tagged by a pair of bracketed descriptive text strings which form an opening tag and a closing tag. So, for example, the data in a data group is incorporated between opening tag <Datagroup2> and a closing tag </Datagroup2>.

Because of the size of the individual descriptive tags and because of the provision of both opening and closing tags for each element of data, the XML metadata schema shown in Figure 1 is verbose. The size of the data file can be reduced as shown in Figure 2a. Thus, Figure 2a shows a text string representation of the schema of Figure 1. In the representation of Figure 2a, the data group, the elements, and the sub-elements are listed in order. This enables closing tags to be generally omitted from the text string representation. As also shown in Figure 2a each bracketed descriptive text string as <Datagroup2> has been replaced by an alphanumeric, alphabetical or numeric opening tag. Thus, the <Datagroup1> tag, for example of the schema of Figure 1 has been replaced by the opening tag 16 MD01. In Figure 2a there is also shown a closing tag 18 (/MD01).

In the example shown in Figure 2a, alphanumeric tags as 16 are used to identify data groups. The alphanumeric tags comprise initials and a sequentially increasing number. In the example illustrated the metadata in the first data group 10 has the opening tag MD01 and the sequence MDnn is sequentially increasing. The elements and sub-elements are identified by numeric tags and as is apparent from Figure 2a, sequentially increasing unique numbers are used to identify individual elements 12 and individual sub-elements 14.

It is important that the number of variable states available for each tag is greater than the number used to allow for extensibility. Leading zeros can be utilised if necessary.

It would be possible to use any format of alphanumeric, alphabetical and numeric tags to reduce the size of the data file. However, the use of a different format tag for data groups and for elements within each group gives robustness to the scheme as it provides the hierarchical structure. Of course, identical referencing between the metadata elements and the encoding tags must be used in a client parser and in a server encoder so that a consistently maintained schema is developed and used at both ends of the system.

Figure 2b exemplifies data 20 which is to be used to partially populate the text string representation of Figure 2a. Together Figures 2a and 2b illustrate a partially populated text string representation of the markup language schema of Figure 1.

Figure 3 shows the formation of the text string representation of Figures 2a and 2b into a message string for transmission. It will be seen here that in the message string, the only pair of start and end tags are SchemaName 22 and /SchemaName 24 which identify the start and end of the encoded data. Otherwise closing tags are not used and are not required. The provision of an opening tag implies the end of the preceding group or element.

Run time encoding and decoding algorithms generally require the parser to remain in sync with the data stream and for the data to be encoded in a specific order defined by the schema. However, in this case, the alphanumeric

tag, for example, MD02 also allows complete metadata data groups (defined by the MDnn – /MDnn tags) to be omitted from an encoded sequence. Also elements at any level may be omitted provided that their child elements are not included, and provided that overall the parent-child relationship is maintained in the encoding and decoding.

The closing tags shown in Figure 2a in brackets, for example, /MD02, would be implied by the beginning of the next data grouping of the metadata or by the end of the schema and would not be carried in the encoded data.

Space separators 26 are necessary between encoded tags and actual data in the encoded metadata to prevent any possible problems in the parser. Spaces are therefore not allowed in any of the tag names or strings to be encoded unless they use standard escape sequences.

It will be seen that in the example of the text string representation shown in Figures 2a and 2b, the data group MD03, for example, has no sub-elements and no data. Groups of elements without data as the data group MD03 and the elements 4 and 10 may be omitted from the string as also indicated by Figure 3.

Figure 4 illustrates processes for transmitting and receiving data. In Figure 4 data 30 is encoded by an encoder 32 to provide a data file based on a markup language schema. This schema is stored by a store 34 for use by the encoder. The encoded data file is then encapsulated for transportation as indicated at 36 and may be transmitted by any appropriate media. At a receiver the transmitted data file is de-encapsulated as indicated at 38, and the encoded data file is then applied to a decoder 40 for decoding in accordance with the same schema 42. In this way the original data 30 is obtained.

With the present invention, the encoder 32 is controlled to encode the data not only in accordance with the markup language schema but also in accordance with the invention such that the resulting data file is reduced in size. The transmitted data file is, therefore, smaller than previously.

It will be appreciated that amendments to and variations of the embodiments specifically described and illustrated may be made within the scope of this application as set out in the accompanying claims.

CLAIMS

1. A text string representation of a markup language schema which defines a structure for carrying information and metadata, which information and metadata is identified within the schema by appropriate tags, wherein elements of information and metadata are not identified by bracketed descriptive text strings provided as both opening and closing tags but by alphanumeric, alphabetical or numeric opening tags.

2. A text string representation as claimed in Claim 1, wherein, in the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements, the text string representation listing the data group and its elements and sub-elements in order, and wherein the alphanumeric, alphabetical or numerical opening tag identifying each data group, element or sub-element additionally implies the close of the preceding data group, element or sub-element

3. A text string representation as claimed in Claim 1 or Claim 2, wherein the only closing tags provided identify the end of a schema.

4. A text string representation as claimed in any preceding claim, wherein, in the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements, the text string representation listing the data group and its elements and sub-elements in order, and wherein alphanumeric tags are used to identify data groups and numeric tags are used to identify elements and sub-elements

5. A text string representation as claimed in Claim 4, wherein alphanumeric tags identifying data groups comprise initials and a sequentially increasing number.

6. A text string representation as claimed in any preceding claim, wherein, in the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the

data group, and sub-elements within the elements, the text string representation listing the data group and its elements and sub-elements in order, and wherein sequentially increasing unique numbers are used to identify individual elements and sub-elements.

7 A text string representation as claimed in any preceding claim, wherein a space separator is provided between an opening tag and the identified data.

8. A text string representation as claimed in any preceding claim, wherein the number of available states of each alphanumeric, alphabetical or numeric opening tag must exceed the number of states to be used.

9. A method of encoding a data file based on a markup language schema to reduce the file size, where the schema defines a structure for carrying information and metadata which is identified within the schema by appropriate tags, where the elements of information and metadata in the unconverted schema are identified by bracketed descriptive text strings provided in pairs as opening and closing tags, the method comprising omitting the closing tags.

10. A method of encoding a data file based on a markup language schema as claimed in Claim 9, wherein the only closing tag retained is a closing tag identifying the end of a schema.

11. A method of encoding a data file based on a markup language schema as claimed in Claim 9 or Claim 10, comprising replacing the bracketed descriptive opening tags with alphanumeric, alphabetical or numeric opening tags.

12. A method of encoding a data file based on a markup language schema as claimed in any of Claims 9 to 11, wherein, in the markup language schema, the information and metadata is arranged in a hierarchical structure comprising at least one data group, elements within the data group, and sub-elements within the elements, the method comprising listing the data group and its elements and sub-elements in order, and comprising replacing the bracketed descriptive opening tags identifying data groups with alphanumeric tags, and

replacing the bracketed descriptive opening tags identifying elements and sub-elements with numeric tags.

13. A method of encoding a data file based on a markup language schema as claimed in any of Claims 9 to 12, comprising incorporating a space separator between an opening tag and the identified data.

14. A method of encoding a data file based on a markup language schema, as claimed in any of Claims 9 to 13, further comprising forming the encoded data file into a string for transmission.

15. A method of encoding a data file based on a markup language schema as claimed in Claim 14, further comprising compressing the formed string.

16. A method of encoding a data file based on a markup language schema as claimed in any of Claims 9 to 15, wherein the encoding is undertaken by a client parser and a server encoder.

17. Apparatus for encoding a data file based on a markup language schema to reduce the file size, where the schema defines a structure for carrying information and metadata which is identified within the structure by appropriate tags, where the elements of information and metadata in the schema are identified by bracketed descriptive text strings provided in pairs as opening and closing tags, the apparatus comprising a client parser to identify the defined schema and the opening and closing tags, and a server encoder to replace the identified opening and closing tags with alphanumeric, alphabetical, or numeric opening tags.

18. A process for transmitting data, the process comprising encoding the data to form a data file based on a markup language schema, encapsulating the data file for transport, and transmitting the encapsulated data file, wherein the process further comprises reducing the file size of the data file before it is encapsulated using a method as claimed in any of Claims 9 to 16.

19. A process for receiving data, the process comprising receiving and de-encapsulating an encapsulated and encoded data file to retrieve the encoded

data file, and decoding the data file to obtain the data, wherein the data file had been encoded based on a markup language schema and reduced in size using a method as claimed in any of Claims 9 to 16, and wherein the data is decoded based on the markup language schema.

Intellectual Property Office

For Creativity and Innovation

Application No:	GB0714906.5	**Examiner:**	Kalim Yasseen
Claims searched:	1-8, 17	**Date of search:**	16 November 2007

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	at least 1, 17	US6635088 B1 (IBM) see whole document especially col. 11 lines 59-66, col. 12 lines 45-60
X	at least 1, 17	Computer Networks 33 (2000) 747-765 "Millau: an encoding format for efficient representation and exchange of XML over the Web"; GIRARDOT M; SUNDARESAN N; see page 764 lines 21-37, XP004304805

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{X} :

Worldwide search of patent documents classified in the following areas of the IPC

H03M

The following online and other databases have been used in the preparation of this search report

Online: EPODOC, FULLTEXT, INSPEC, NPL, WPI, XPESP

International Classification:

Subclass	Subgroup	Valid From
H03M	0007/30	01/01/2006
G06F	0017/30	01/01/2006