Description

"Dynamic method for generating XML documents from a database"

[001] TECHNICAL SCOPE OF THE INVENTION

[002] This invention relates generally to databases generated by computers, and more particularly to the automatic generation of documents of the type using markup language XML, "eXtensible Markup Language", when interrogating databases.

[003] STATE OF THE ART

[004] Databases generated directly from computers or through a network have been used for a very long time by numerous companies, administrations and all kinds of commercial or other organisations. Databases are entities wherein it is possible to store data in a structured manner with the least possible redundancy. The major advantage of using databases is therefore the fact that they can be accessed by a number of different users simultaneously.

[005] For example, there are databases that contain all the timetables and the status of reservations of the flights of all the airlines throughout the world, or all the offers of international hotel chains. Travel agencies may then access this information from anywhere in the world, particularly now that the Internet allows all private or public networks to be interconnected. Therefore the concept of a database is generally linked to that of a network in order to be able to combine and share this information.

[006] In order to control the data, as well as the users, a database management system (DBMS) is required. The DBMS is an assembly of software applications that enable databases to be managed, i.e. to allow simple access to the data, authorise access to the information for multiple users, and to handle the data present in the database (entry, deletion, modification).

[007] In most cases databases are constructed according to a so-called relational model, which organises the data in the form of tables. The data are handled according to the mathematical concept based on the theory of assemblies, i.e. relational algebra, which consists of an assembly of formal operations on the relations. The relational operations enable a new relation (table) to be created from

elementary operations on other tables (for example, combination, intersection, or even difference).

[008] A structured language of queries has also been widely adopted. Known under the abbreviation SQL, "Structured Query Language", it is at the same time a data definition language and a data handling and control language. The definition of data is the work of the administrator. Most people who use the SQL language only use the data handling language, which enables them to select the data of interest to them. From the user's point of view the most important command for the data handling language is the SELECT command. Based on relational algebra, it enables data selection operations to be carried out on several relational tables in order to make them available to the user.

[009] With the extraordinary development of the Internet, and the interlinking of public and private networks it has allowed, the exchange of data has been standardised by the adoption of a protocol called HTTP, "Hyper Text Transfer Protocol", and through the development of a communication system known under the name of Web, the "World Wide Web". The World Wide Web allows information to be searched, it allows access to this information and its visualisation using hypertext or HTML, "Hyper Text Markup Language". This language enables hypertext documents to be created that can be displayed by a Web browser.

[010] The HTML language is a special, very simple version of a much more sophisticated standard adopted in the 1980's to describe documents and their formatting, known under the abbreviation SGML, "Standard Generalized Markup Language". The SGML language is a symbolic language which is still used to describe the logical structure of a document, and not its topographical aspect. The principle applied is generic marking, by means of markers, of chapters, paragraphs, preface, introduction and other components of a document. Although the simplicity of the HTML language has contributed largely to its adoption, and was wholly adequate at the beginning of the Web's development, the requirement for more demanding applications quickly demonstrated its limits. Because the simplicity of HTML was achieved at the cost of absence of scope for further development, the

need was felt very soon for a language that offered possibilities of development and adaptation similar to the standard from which HTML was derived, i.e. SGML, but avoiding the complexity of the latter. The response to this was a new standard proposed at the end of 1996 by the organisation entrusted with the task of standardising the Web, this standard being known as XML, "eXtensible Markup Language". Just as SGML, but without its complexity, XML is a metalanguage which enables the HTML language to be extended. In particular, it enables new markers to be defined, for example, for a clearer description of a text, and it enables documents to be fully personalised. The success of XML lies in its ability to describe any data domain thanks to its extensibility. It enables the vocabulary and syntax of the data it contains to be structured and established.

[011] It is therefore not surprising that the XML language has been used since its creation for formatting documents containing data derived from databases, particularly relational databases described above. Data obtained by means of SQL queries.

[012] However, the use of the data obtained, and their inclusion in documents for their display in a window opened by the Web browser of a user, may require major coding work. In the example already quoted above, of databases interrogated by travel agencies, the users, the required level of personalisation of the display may be very high. In fact, the customers of these travel agencies have highly varied requirements. Depending on whether they are holidaymakers or businessmen or – women, the selection of the data and their presentation may be very different in order to facilitate the work of the staff in these agencies, increase their productivity and result in greater customer satisfaction.

[013] The automatic generation of XML documents has therefore attracted particular attention from the developers of these applications. Reference may be made, for example, to a patent issued by the American Patent Office (USPTO) under number US 6,636,845, which describes a method for generating XML documents from a single SQL query "Generating one or more XML documents from a single SQL query". However, the method described lacks flexibility. In particular, it relies

upon a static definition to describe access to the data. Therefore there are as many definitions as types of data to be accessed in the corresponding XML documents.

[014] OBJECT OF THE INVENTION

[015] Embodiments of the invention seek to propose an improved method for generating XML documents from a database.

[016] Embodiments of the invention further seek to be able to generate an XML document from a dynamic definition of access to the data in the database.

[017] Embodiments of the invention further seek to obtain this dynamic definition by using a combination of nested templates described in the database format.

[018] Embodiments of the invention further seek to simply and reduce the number of XML documents to be generated for an application whilst maintaining a high degree of personalisation.

[019] The other objects, characteristics and advantages of this invention will be apparent to the persons skilled in the art on examining the following description and accompanying drawings. It is understood that other advantages may be incorporated.

[020] SUMMARY OF THE INVENTION

[020a] In accordance with a first aspect of the present invention, there is provided a method for automatically generating at least one XML document including data extracted from a source database, said database comprising means for generating said at least one XML document, wherein said means includes an XML generator making use of tables defined in the format and with the language of said database, said method consisting in: defining a plurality of templates in at least one definition table, said plurality of templates including executable instructions by said XML generator, wherein said executable instructions include at least one call instruction for calling any of the templates of said plurality of templates; defining queries associated with each of said templates, wherein said queries are used to extract data from said source database in the form of sets of records; executing said instructions during an execution stage from a file definition included in said at least one definition table,

wherein said execution stage includes a call by at least one of said templates to at least one other template of said plurality of templates in order to be able to generate said at least one XML document including said data deriving from said sets of records.

[020b] In accordance with a second aspect of the present invention, there is provided a system, in particular a generator of XML documents, said system including means adapted to implement the method described in the first aspect.

[020c] In accordance with a third aspect of the present invention, there is provided a medium readable with a computer comprising the instructions of a program executable by said computer, said program implementing the method according to the first aspect.

[021] A method for automatically generating at least one XML document that contains data extracted from a source database is described. The method consists in defining a plurality of templates in the database format; such templates contain executable instructions. Queries associated with each of the templates are also defined. The queries are used to extract data from the source database in the form of sets of records. The instructions of the templates are executed in order to generate at least one XML document that includes the data deriving from the sets of records. The instructions include at least one call instruction for calling any of the other templates. The XML documents consist of elements that include attributes. The method is characterised in that it maintains a current element concept. The method also includes an instruction to create a new element which is assigned to the current element. The new element then in turn becomes the current element. The method also includes an instruction to create a new attribute to the current element, as well as an instruction to close the current element, which consists in reassigning the role of

current element to the parent element of the current element. The method includes a file definition that makes it possible to construct the name of the XML document, to create a root element which becomes the current element, and to specify which of the templates is the root template. Parameters are used to link the templates when the template call instruction is executed or when the file definition is open and the root template is used. Each of the template instructions is applied to each recording of the sets of records. The instructions are interpreted by the XML generator. The source database is a relational database and the queries are queries which obey the rules of the structured query language known as SQL. The templates, queries, instructions, file definition and parameters are contained in the form of tables in at least one relational database and may be included in the source database.

[022] BRIEF DESCRIPTION OF THE FIGURES

[023] The purposes, objects, as well as the characteristics and advantages of the invention will become more apparent from the detailed description of an embodiment of the invention, which is illustrated by the following accompanying drawings, wherein:

[024] FIGURE 1 shows the overall method of implementation of the invention.

[025] FIGURE 2 describes the structure of the definition tables of the XML generator.

[026] FIGURE 3 describes the process which relates to the generation of the files on the basis of the content of the file definition.

[027] FIGURE 4 descries the stage of creating the target XML file.

[028] FIGURE 5 describes the processing of a template.

[029] FIGURE 6 describes the instructions of the templates.

[030] FIGURE 7 is an exemplary embodiment of the invention.

[031] The attached drawings are given by way of example and do not limit the scope of the invention.

[032] DETAILED DESCRIPTION OF THE INVENTION

[033] **Figure 1** shows the overall method of implementation of the invention, which is used for a software application referred to below as VISTA (100), the purpose of

which is to make available to remote users information extracted from a source database (140) by automatically formatting them in a form that can be easily interpreted by the browsers of these users. The database will, for example, contain the offers and status of reservations of international hotel chains. In this example the users are therefore travel agencies, which can then access this information from anywhere in the world from a public or private network via the Internet.

[034] As discussed in the chapter on the state of the art, the XML language is the most suitable for this purpose. Proposed by the main Web standardisation organisation, i.e. W3C, the "World Wide Web Consortium", it has been very widely adopted. VISTA is therefore based on the generation of documents in the form of XML files (110) for displaying their content, thereby making the information sought available to the users. For this purpose the application (100) must read the XML files by means of a syntactic analyser (parser).

The parser is a standard tool developed and sold by software suppliers such as the well known American company the Microsoft Corporation.

Analysis of the target files generates, during the execution of the parser, objects (105) constituting pages that can be displayed by the browser of the application user.

[035] The XML generator (120) used to generate XML files is based on models or templates (the term "templates" is used in the English technical literature on these subjects). They are stored in the definition tables (130) of the XML generator in the format and in the language of the relational database used. The templates can be nested to increase their possibility of being re-used. The templates according to the invention are sets of instructions that can be interpreted by the XML generator and can be re-used in their context. The context of a template applies to a set of records, or record set, according to the English technical terminology, i.e. to a partial view of the database obtained by means of a specific query issued to the latter. The queries conform to the structured query language (SQL) as previously discussed. According to the invention the queries are defined and stored in the database (135) in the form of objects which each provide access to a particular set of records. The query objects may contain parameters. The parameters then serve to modify the query. For

example, in a database which lists all the books in a library, a query may be defined that enables the books to be searched for by the name of a particular author and thus includes an "author" parameter. For a specific search, the author parameter may then be replaced by the name of a specific author. The corresponding set of records, which will be returned by the database in response to the query, will consist of the list of books by that author held in the library.

[036] If XML is the most suitable means of distributing the data to the users, a relational database (140) is ideal for storing the data relating to the product. The use of a well structured database guarantees easy and effective maintenance of the information it contains and prevents redundancies, in particular. The database (140) is therefore the essential source of the data that will be included in the XML files generated for users of the application (100). The method of the invention takes advantage, in particular, of the use of the structured query language SQL already mentioned. The use of templates, in the format of the database, which enables the target XML documents to be generated, does not require the distribution to the users, at the time they connect with their browsers, of the software resources that they would otherwise need to access the source database directly. This would necessarily increase the time required to access the information sought, block the network and make it very complicated to make available the information sought. In fact, although the structure and organisation of the database tend to guarantee its efficiency and avoid redundancies, the display of the information to the user meets other criteria. The matching of the structure of the data in the database to the data required by the user can just as well be carried out at the source, as recommended by the invention, by the XML generator (120), which relies upon the templates of the definition tables (110) and the associated query definitions (135) to create a small number of target XML files (110) containing sufficient information for the user.

[037] The definition tables, the templates and queries according to the invention may be contained in a database that is separate from the source database (140), as shown diagrammatically in Figure 1, but could just as well form part of a unique database (150) without problem.

[038] The purpose of the invention is therefore to generate XML documents (110) from a single source (140), a relational database, managed by software applications which constitute the actual standards, of which there are numerous well known commercial variants, for example those known under the names "Oracle", "IBM DB2" or "Microsoft Access", products developed and marketed by their respective companies. The management and integrity of the data themselves are therefore provided by the use of a standard product that has been widely tested and used. Therefore, by including the rules for generating and testing XML documents in the database itself in the format of the latter, numerous advantages may be derived from this. In particular, the development time is shortened. Making changes in a database and automatically regenerating all the XML files of a given application, such as VISTA (100), therefore takes much less time than having to update and correct a specific code. Moreover, this can be carried out directly by the professionals responsible for developing the application itself, VISTA in this example, whereas the development of a specific software always requires expensive coding work by highly specialised programmers.

[039] **Figure 2** describes the structure of the "definition tables" of the XML generator. Each table contains:

[040]

Table 1

File Definition (200)	A file definition is required to create the target XML file. The file definition includes the following information: - the name of the XML file to be created - the root template to be called - the associated query for finding the suitable set of records in the database - the root element to be created (optional) - the parameters (as defined below)

Template (220)	A template contains executable instructions as defined below. A template may call another template (225).
Instructions (230)	The execution instructions of a template are of several types which serve to: - create an XML element - create an XML attribute - close an XML element - call a template (parameters may be specified)
Parameters (210, 240)	The parameters act as links between the set of records of the calling template and that of the called template, and a distinction must be made between: Parameter definition: The parameters stored in the templates (240) and file definition (210). The definition includes a name and a field name of the current record set, which will contain the value of the parameter from the following group of parameters. Group of parameters: A temporary group of parameters of the set of records deriving from the database. Parameters of queries: For finding the suitable set of records in the database.

[041] The file definition (200), which makes use of parameters (210), contains the definition of the particular query to be applied to the database to find the appropriate set of records. The file definition references a template (220) which makes use of instructions (230). When the instruction is of the type which calls another template (225), parameters must then be passed to the called template (240).

[042] The elements and attributes of the XML document mentioned in the above table form an integral part of the XML language and are concepts well known to the persons skilled in the art. Generally an XML document will include one or more

elements bounded by start and end markers in the form <xxxElement> and </Element> respectively. Each element may then have one or more attributes, each having a name and value which are specified in the template instructions.

[043] **Figure 3** describes the process of generating files on the basis of the content of the "file definition" described above and forming part of a "table of definitions of the XML generator".

[044] After the file definition (300) is opened for each record (314) for this definition, it is tested (320) whether there is an associated "query definition". If this is the case (322), the corresponding query is executed. The "set of records" obtained from the database is then opened (330). There will as many target files created (360) as there are records in the record set. All the records in the record set are scrolled through (350). When the last has been processed (334) the process passes to the next record of the file definition (370).

[045] If, when testing for the presence of an associated query definition (320), the reply is negative (324), a single target file is then created (380) before proceeding to the next record of the file definition. If there are no further records the process is completed (312).

[046] The "target file" creation stage (380) and similar stages for creating the"target XML file" (360) are described in **Figure 4.**

[047] The creation of the target file first of all involves constructing (400) a file name. If, as seen previously, the same "file definition" is used to create multiple target files, the construction of the name results from the concatenation:

- of a prefix deriving from the file definition
- of a dynamic part deriving from the current record of the set of records
- of a suffix deriving from the file definition

[048] However, if there is no associated query definition, as has just been seen, a single file must be created. In this case only the prefix and suffix are used, there is no dynamic part.

[049] After this the target XML file is effectively created (410) in the form of an object representing the structure of the XML document and conforming to the DOM specification (Document Object Model) issued by the Web standardisation organisation, W3C, already mentioned.

[050] If a root element appears in the file definition (422), it is created (430). After this, if parameters are present in the file definition (442), these are located (440) for passing to the template (444).

[051] The processing of the template (450), which may be a root template or a nested template, i.e. called by another template, is described in more detail in **Figure 5.**

[052] The parameters defined by the calling template (or the file definition) must be matched to those of the template called. For this purpose a temporary "group of parameters" is constructed (500) on the basis of the definition of the parameters. When this is done the appropriate values are obtained form the current record of the "set of records".

[053] The stage (500) is broken down in detail in the right section of Figure 5. Each of the parameters is found by its name (501) as well as the field defined by the parameter (502). The value of the field is obtained from the "set of records" (503), which enables the value of the parameter to be fixed (504). The process is repeated (506) as long as there are parameters (505) to be processed.

[054] The called template is then found (510). The parameters of the associated "query definition" must be matched to those of the calling template before execution on the database in order to obtain the appropriate "set of records" (520).

The parameters are applied exhaustively to each of the records in the "set of records" (530) and for each of the instructions (540) until there are no longer any instructions to be processed (560) or a record in the "set of records" (550).

[055] **Figure 6** describes in more detail the instruction processing stage. When the instructions are executed the system maintains a current element concept. It is this element that is the target of each of the instructions described below. If there is no current element the root or first element of the template is used.

[056] The instructions are of several types: creation of an element (600), creation of an attribute (610), closing of an element (620) and calling a template (630). In the first two cases it is first necessary to obtain (602) the value of the field specified in the current "set of records". In the last case it is first necessary to obtain (632) the parameters to be passed to the called template.

[057] An instruction may therefore have to create an XML element (600) bearing the name constructed as previously described. The new element is then assigned to the current element (605), which therefore becomes the parent of the new child element. Optionally, a value may be inserted in the new element and this value must correspond to the field specified in the "set of records".

[058] An instruction may have to assign an attribute (610) to an XML element. The attribute is created for the current element (615). As above, a value may optionally be fixed to this attribute, and this value must correspond to a specific field of the record set.

[059] An instruction may close an element (620), which amounts to reassigning the role of current element (625) to the parent element.

[060] Finally, an instruction may call a template (630) by the name previously specified. For processing the called template (635), parameters will be passed from the set of records of the current template to that of the called template.

[061] **Figure 7** illustrates, by means of a particular example, the method according to the invention for generating a target XML file from a relational database. In this example, as is almost exclusively always the case in an environment where software products are being developed which are to be marketed throughout the world by multinational teams, all the technical terms of the code are defined in English. However, this should not prevent a clear understanding of the following description. When necessary the terms are either explained or they may easily be considered to be a sequence of letters and numbers, with no particular significance, even though attempts are made to assign some meaning to them for obvious mnemonic and practical reasons.

[062] The creation of an XML file begins with the opening (700) of the "file definition", already described, which takes the form of a table (705) in the database. With the information contained in this table it is then possible to create an XML file, with the specified name (codesList_CAR.xml), containing a first empty element (codes_list). This element becomes the current element. The file definition contains the name of the root template (710) which has been used.

[063] The above template is then processed (720). The first stage consists in executing the query (722) defined for the template (CarCodesList), on the database, in order to obtain the corresponding set of records (730). The instructions of the template (710), 5 in this example, are applied to each of the records in the set of records thus obtained.

[064] This starts with the creation (724) of a child element (list) of the current element previously created. This element in turn becomes the current element. The next two instructions, indexes 2 and 3, of the template (710), are then processed. They serve to create (726) two attributes (Name, Area), each corresponding to a Field Name specified in the template. The current element (list) of the XML file (728) then includes these attributes.

[065] The next instruction, index 4 of the template (710), makes use of another template (CodesListItem), for which a parameter is defined (715). The called template (740) also has a query definition (CodesListItem), which enables the corresponding set of records (750) in the database to be obtained after the parameter of the calling template (715) has enabled the field (742) to be found for effectively executing this query.

[066] The called template is processed in a similar manner to that already described above for enriching the XML file (760).

Claims

[001] A method for automatically generating at least one XML document including data extracted from a source database, said database comprising means for generating said at least one XML document, wherein said means includes an XML generator making use of tables defined in the format and with the language of said database, said method consisting in: defining a plurality of templates in at least one definition table, said plurality of templates including executable instructions by said XML generator, wherein said executable instructions include at least one call instruction for calling any of the templates of said plurality of templates; defining queries associated with each of said templates, wherein said queries are used to extract data from said source database in the form of sets of records; executing said instructions during an execution stage from a file definition included in said at least one definition table, wherein said execution stage includes a call by at least one of said templates to at least one other template of said plurality of templates in order to be able to generate said at least one XML document including said data deriving from said sets of records.

[002] The method according to Claim 1, wherein said at least one XML document comprises elements which are able to include attributes, said method characterised in that said execution stage includes a stage for creating said elements and their said attributes.

[003] The method according to Claim 1 or 2, wherein said file definition makes it possible: to construct the name of said XML document; to create a target file for said XML document; to create a root element, said root element to which is assigned the function of current element; to process a root template capable of calling any one of the templates of said plurality of templates.

[004] The method according to Claim 3, including at least: an instruction for creating a new child element of said current element, to which said new

element is assigned in turn the function of said current element; an instruction for creating a new attribute, said instruction creates said new attribute for said current element; an instruction for closing said current element, said instruction consisting in reassigning the function of current element to the parent element of said current element.

[005] The method according to any of Claims 3 or 4, including parameters, said parameters serving to link said templates when said template call instruction is executed, or when said file definition is opened and use is made of said root template.

[006] The method according to any of the preceding claims, wherein each of said instructions of said templates is applied to each record of said sets of records.

[007] The method according to Claim 6, wherein said instructions are interpreted by the XML generator.

[008] The method according to any of the preceding claims, wherein said source database is a relational database, and said queries are queries that obey the rules of the structured query language known as SQL.

[009] A system, in particular a generator of XML documents, said system including means adapted to implement the method described in any of the preceding claims.

[010] A medium readable with a computer comprising the instructions of a program executable by said computer, said program implementing the method according to any of Claims 1 to 8.