Tarek Hoteit
Knowledgeview Ltd
Tarek.Hoteit@knowledgeview.co.uk

Wednesday, April 18, 2001

Reverse XML

Dear Sirs,

The subject of this paper is to present a concept in which I took the liberty to name it, "Reverse XML". The audience of this concept is targeted towards news agencies, newspapers and any related companies that are interested in content syndication with one another without resorting to any modifications of their current system and its corresponding workflow.

For simplicity we shall assume the following:

a) each XML file includes a set of common elements used by all newspapers and agencies. Types of such representations could range from a couple of elements (author, title and body) to a more complex one such as NITF or NewsML;
b) each source feed provided by the client is written in an ASCII text file, with a systematic data distinction within the data itself. Such type of representation does not have to follow a certain international standard but should confirm to the standards that its application is using.

The project aims in providing the following distinctive features:

1. Provide conversion capabilities for customer source feeds into XML standards.
2. Provide a type of "reverse" approach where any XML file could be converted back into any of those clients source fields.

In one part of some customer work-flow cycle, articles are being reproduced by news receivers or by employees themselves via an ASCII text file with a certain type of distinction of each block of data (author, title, etc.) Such articles, and just before the "lay out" stage, may be considered as "raw tagged files". The term "raw" defines the fact that the news feed is not published (layout is not applied yet). The term "tagged" defines the fact that the file has some type of data distinction between the data itself, whether delimiters are being used, or systematic sequence of data is applied, or "**any**" other forms of data distinction. We shall term "raw tagged files" as source data .and the structure of the source data to be source format.

You must have noticed that in the previous paragraph, the word "any" typed in bold. This is intentional because the word "any" would be the focal point for the entire project.

The mere fact that we intend to convert one customer's source format into XML format and maybe then back to the original source format of the customer himself or any other customer (reverse approach), then the term "any" should be clearly represented and considered seriously. If the term "any" could not be made possible to exist than "Reverse XML" would not be universally (within our scope and our assumptions-see the introduction) applicable and therefore limited in its efficiency and features.

How It Works?

We will explain the procedure starting with an example. Suppose two different newspapers use different source format for their source data within their application, whether via some tagging techniques or pattern techniques (please recall that a source format is one different method of representing the different types of data –author, title, etc. within the news feeds themselves).

Company1:
A sample of source data represented in Company 1 source format may be as the following:

```
Indiana Jones
***
Diary of my adventures at the Temple of Doom
***
Woke up one morning blah blah blah blah blah blah
$$$
My picture indy.jpg
```

Fig 1000 – sample source data for Company 1

The above sample file shows Company 1 source format where field values are separated by delimiters. In general terms the source format of Company 1 is as follows:

```
{author}
delimiter
{title}
delimiter
{body}
different delimiter
{captions}
```

Fig 1001 – source format for Company 1

where the different delimiter (see Fig1000: $$$ instead of ***) would mean that the next field is optional.

With such source format, some pattern exists where the first block of data and till the first delimiter are values for field "author", and then the next block and till the next delimiter are values for field "title", and so on so forth.

We shall assume that all source data for Company1 are of the same pattern, as above, (with or without the optional fields) and therefore we could be able to derive a consistent pattern. If the pattern is broken, then the corresponding source data would not even be valid for the company's application, right?!?

Company2:
A sample of source data represented in Company 2 source format may be as the following:

```
(Author: Spideman
(Title: One thousand and one tricks in climbing walls.
(Body: ping pong ping pong ping pong ping pong)
(Caption: My picture squashed.jpg)
```

Fig 1003 – sample source data for Company 2

The above sample files is one type of source data where field values are separated by some tag representation. Company 2 would giggle a little bit and tell you that the above file is structured with the following:

```
(Field Name: Field Value)
```

Fig 1004 – source format for Company 2

where only field named "Caption" is optional.
Similar to Company 1, we shall assume that all source data for Company2 are of the same source format, as above.
Company 1 and Company 2 are just two examples of supposedly infinite number of companies with an infinite number of different source formats. Adding to that the number of combinations of relationships between Company 1 and Company ∞ may be infinite. Therefore one **common middle ground** should be assumed where all Company x would dump in **copies** of its source data. Why copies? Simply our whole assumptions from the start that Company X would want to give data to Company Y (where Company Y could be anything from another newspaper to some portal) without any modifications to its system. Such "common middle ground" could be any kind of format and I have also taken the liberty, for the second time, to choose this format as XML. Why?

1. A client would have a copy of her data converted into a universally standard format, without any modifications of her system, for reasons that are beyond the scope of this paper.
2. XML is universally identified with powerful features, such as well-form, validity checking and transformation capabilities.

FAQ:
Are we using to transformation capabilities of XML to convert the infinite different non-XML source formats for companies A-Z? No, unless the orignal source format is written in XML and such would be exceptions that the application should handle.
If in some way or another we were able to convert the source format of Company X into an XML format, is that it? Company Y does not use XML in its source data, what shall it do? That is the catch here. If the source format of Company X was converted to XML format, then "Reverse XML" should be capable of converting the XML file back into the source format of Company Y or any one from Companies A-Z!

Fig 1005 – Frequently Asked Questions

A content-representation language will be established for "Reverse XML" where Company X could describe the source format of its source data.
A file written with our content-representation language would be named: "**content key**" file. This special file would be used to convert the source format of Company X into XML as well as from XML to the original source format, similar to cryptography techniques (a key to encode and decode).
"Reverse XML" would NOT store any of the Company X source data but would only store the "content key", that would used to convert its source data into XML format. However it is the responsibility of Company X to make modifications to its "content key" when appropriate.

Going back to example for Company 1 and Company 2. Company 1 would write its own "content key" file for its source format, based on the content-representation language. With "Reverse XML", the "content key" file of Company 1 would be used to map the source data into an XML version. These XML files would be returned to Company 1 in order to do whatever it wants with them: portal sites, business with Company 2, etc. Suppose these data are to be handed in to Company 2 who in turn has a different "content key". With the XML version of Company 1's data and the "content key" file of Company 2, the data would be reversibly translated from XML format into the source format of Company 2.

Conclusion

Company X, like a newspaper company, would be interested at some point in providing XML versions and Company A-Z versions of their data without any technical modifications for their system. With "Reverse XML", we are ensuring that companies could focus on their work where data format conversion between one another is done by "Reverse XML." The application may be established as a **free** service where the following could be our revenue:

1. Our database would include all companies' source formats, where our content-representation language that should handle all of these formats may create a data bridge between all those companies whose applications are not XML-friendly yet.
2. Having said point "1", our database would be also valuable because we will be able to market-focus our products that maybe of interest to these companies.

Note: "Reverse XML" may be free to use and our company may provide as a paid service the option to write the client's "content key" files.

T.H. (4/18/2001)