Event Based Content Editor
(Concrete Implementation of the eventBased Data Philosophy)

People use the content (in general text) as the main output of their work. I want to provide an ability of storing information about every procedure which the human has done on this output. In such a way that when a procedure binds pieces of text in some relations, this binding to be technologically presented and when a piece of text is edited, the user to be aware (by warning) of the relations of the edited text with other text(s) pieces. I want to ensure not breakable logic relations between pieces of data when future editing is applied.

Garo Garabedyan
Independent researcher,
Student at Technical University- Sofia, Bulgaria
garabedyan@gmail.com

1. Used expressions and their meanings in this paper:

1.1 Text/ Content

Text refers to every group of words logically forming connected statements. Text in its nature has an object of representation. Valuable content is always dynamic, it changes when there is a change in the purpose of the document, in documents which relate to this one and in many other consequences which reason the need of change of the content.

In technology aspect content is symbols, pictures, tables, etc.

1.2 Content Editor

Content editor generalizes a group of applications which deals with content. These are applications enabling editing of content and this way creating revisions of it.

1.3 Connections between parts of content

In any text we found connections between its parts and parts of other texts. Of its own genesis this connections are dynamic as the nature of the content.

Relations are logical and lexical according to their nature of genesis. When a relation breaks, this causes a logical or lexical mistake.

This logical relations count is not discrete. Often we can't count them. We as editors of content are taught to capture many of this relations, but some of them are bind with the concrete content and its aim. The letter relations are too heavy and sometimes with a couple of readings aren't enough to recognize such a broken relation.

1.4 Part of Content/ Text

Text which can be marked in the editor. Copied and pasted in the same or other document.

1.5 Connection between Parts of Text

Group of texts connected in a connection. Minimal count of this group is 1 /one/ (in order to enable user to start some connection and in future to find text appropriate to be added to the connection), maximal count of this group is not set at this moment. Every connection can have text describing the connection, this text is called down as note to the connection. It is possible to not enable

the note to the connection to be text (see the definition of text up in this paper) but only group of symbols.

2. Description of the eventBased Content Editor:

2.1 Event Based Text Editor Catching Changes and Its Related Parts in Order to Provide Applying of the Change to the all Related in any Manner Pieces

This is a text editor with a functionality presented to the author(s) to declare connections between parts of the text and write some notes describing the declared connection. Every time when some part of text in this editor is changed and this change affects some part which is part of connection, to the user (in future we can address the output of this functionality not only to human but to a machine) is presented information about the text parts in the connection (which presentation allows the user to see the whole part of text in the connection) and notes written about the connection.

One part of text can be part of many different connections. Every part of text can contain many parts of text which are parts of connections.

The user can view/edit the connections for every part of text by using a viewer integrated in the text editor.

2.2 Advance functionalities

Adding special object treated as part of text which is not printed, but in its own is a macro, which outputs texts and as every part of text this one can attend connections. This objects are supposed to be checked by the editor by running background process in the operating system and when the output of the object (macro) is changed the editor to present a warning message containing the two outputs of the object (present and past) and the parts of the text in the connection(s) and the notes to the connection(s) plus fast way to open the document and edit this parts of text.

The macros have to store in itself the last its output.

Note: It is not an aim of this paper to work on the idea to present the output of the special object (macro) directly to the text of the document.

2.3 Possible implementation

The Event Based Text Editor is written in form of macro and added to the general templates of any Text Editor providing macros. The user is supposed to be provided with a button for accessing the viewer of connections and creating connections tool. Every connection of its own is stored in other macro (one for the all connections in a document) which is not placed in the template but without stored connections in it.

This implementation in its own is multi platform and saves the connections through different computers when they are in the same document.

It is not sure what will happen when part of text (which is part of a connection) is copied and pasted in the same or in other document.

One possible business logic is: Connections are specific in the context of the whole document and it's not right to automate the connections, but only prompting for this ability at every copying. Argument (in example): it's possible the user to try to copy part of text which is part of connection which connection has in itself parts of text which are not in the part of copying text. It's possible this texts to be part of connections with other parts of text not in the copying text.

Q: Connections declared by the user are not in deep about lexical and all connections (in general) but only about the most important ones?

Q: It's important to keep only the first level of connection?

Example:

<1....1> <2.....2> <3.....3>

As its shown it is possible to be selected parts of parts of connections. It is not a good idea to try to automate the copying of connections in every case.

<1.....<2.....1>....2>

Here is an example showing the specific nature of part of texts, it's not a good idea to use XML style of presenting the connections.

2.4 Possible additional tools

Tool enabling finding differences between two documents by using their connections as a start point. In general searching not for connections changes but for the whole document.

Tool working in background of the operating system and from a list of declared files to take the objects (macros) from them which are part of some connections and trace check for differences of the output.

2.5 Embedded RDF in XML (XHTML) implementation can enable referencing connections from different document and even from different sources

http://www.w3.org/2006/07/SWD/RDFa/primer/

When a connection contains parts of text from different sources the system (according to me) should create RDF names referring to the texts attending the connection(s) and place this content in an RDF file attached to the content. It is not necessary to program all parts of text to automatically create RDF names, but I don't like the idea to create RDF name referral when the part of text is connected not only with parts of text from the current document but with other part(s) of text from different document. It's nice the connections about pieces of only one document to be referred to the RDF file attached to the document, too. This connections' references should related to the parts' of text references and this way forming RDF model of the connections and the parts of text. This way machine can be able to find this connections through parts of RDF requests (in order to take the all part of text which is a member of a connection the machine should use a new protocol/ trying to translate the document to an HTML file with a RDF) but not through a completely new protocol.

2.6 Tracing revision

http://garabedyan.wordpress.com/2007/11/18/ajax-wiki-editing/

In the above post I have explained that editing documents is just Editing or Adding content. In this way of thinking by using JS in Web Presentation Layer it can be easy to trace which is edited and which added.

http://garabedyan.wordpress.com/2008/01/27/google-offer-of-cutting-letter-count-in-words-with-the-aim-to-save-memory/