Compus: Visualization and Analysis of Structured Documents for Understanding Social Life in the 16th Century

Jean-Daniel Fekete
Ecole des Mines de Nantes
4, rue Alfred Kastler, La Chantrerie
44307 Nantes Cedex, France
Tel: 33 2 51 85 82 08
E-mail: Jean-Daniel.Fekete@emn.fr

Nicole Dufournaud
Université de Nantes, Faculté des Lettres
Chemin de la Censive du Tertre
44312 Nantes Cedex 3, France
E-mail: Nicole.Dufournaud@humana.univ-nantes

ABSTRACT

This article describes the Compus visualization system that assists in the exploration and analysis of structured document corpora encoded in XML. Compus has been developed for and applied to a corpus of 100 French manuscript letters of the 16th century, transcribed and encoded for scholarly analysis using the recommendations of the Text Encoding Initiative. By providing a synoptic visualization of a corpus and allowing for dynamic queries and structural transformations, Compus assists researchers in finding regularities or discrepancies, leading to a higher level analysis of historic source. Compus can be used with other richly encoded text corpora as well.

KEYWORDS: Information Visualization, Visual Data Mining, Structured Documents, SGML, TEI, XML, XSL, Computers and the Humanities, History.

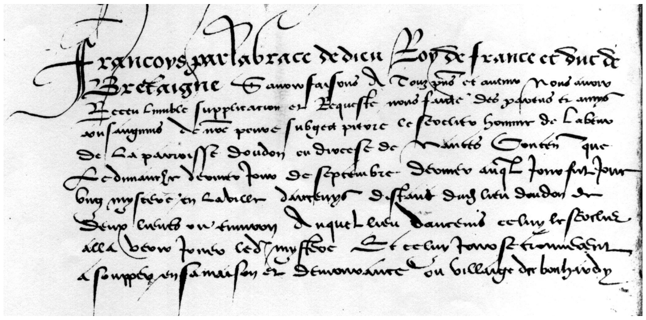
INTRODUCTION

Exploration and analysis of historical textual corpora is difficult. Researchers are faced with large numbers of documents that have never been studied or even read, seeking to "interesting" historic facts that, once structured will confirm or contradict existing hypothesis. Researchers currently rely on their notes, files and their own memories. Notes and files are hard to search, update and share among researchers. Memory is not thorough and prone to error. Will it change in the 21st century?

The field of "computers and the humanities" has been very active in the last decade, recently summarized in [14]. Most of the work has focused on data formats and conventions for interchange. Several digital libraries containing textual archives for research in humanities are now accessible from the Internet, such as [19, 5]. XML [4, 7] promises to be the next archival format for textual material in digital libraries [11], however, very few tools exist to support exploration and analysis using XML encoded documents [21, 16, 17, 18].

This article describes *Compus*, a system designed to support and improve the process of finding regularities and discrepancies out of a homogeneous corpus of encoded textual documents. Developed in close collaboration with French Social History researchers, Compus visualizes a corpus of documents encoded in XML and supports refinement through dynamic queries [1] and structural transformations.

The next section describes our motivations for designing Compus and some details about the encoding of texts using the recommendations from the Text Encoding Initiative [23]. The third section presents Compus using examples from our corpus. The fourth compares Compus with related work. We conclude with a discussion of the use of Compus for research in other fields of humanities such as literature or linguistics before concluding.



Francoys par la grace de dieu Roy de france et duc de Bretaigne [illegible]

Figure 1: Beginning of a "Lettre de Rémission" of 1520

MOTIVATION

This work started as an experiment in using structured documents and the recommendations of the Text Encoding Initiative for conducting research in early modern history [8]. We had to explore a corpus of letters of Clemency (*Lettres de Rémission*) kept in the regional archives of Nantes. These letters are manuscripts from the 16th century, administrative handwritten copies of letters of Clemency usually granting pardon from the King or the Queen. These early modern manuscripts have not been thoroughly studied and have not been transcribed. The historical work consisted of transcribing 100 letters, as in Figure 1, issued between 1531-1532 and studying

Digital Libraries, San Antonio, TX.

"interesting" details of the social life at that time, just before Brittany was linked with France.

Historians are interested in these letters because they describe details of everyday life as well as habits, customs and social relations. The topics they address vary from the architecture of the city of Rennes' jail, to the vocabulary of love and hate. Their extensive analysis is almost impossible but, by reading a large sample, historians are able to extract details that match their research domains and use the letters to correlate their theories or find counter examples of established theories.

In general, given a corpus, a historian will first search for a list of topics of interest. A topic is of interest if the historian is familiar with it and either several documents address it or a document presents new or unexpected facts about it. The former case needs further analysis whereas the latter is already an "interesting historical finding" on its own. In our corpus, criminality is the main recurrent topic, but the social role of women also appears in several letters, as well as the use of weapons. To further analyze a recurrent topic, the historian uses different methods such as counting and statistics, correlation with other topics, linguistic analysis (i.e. what words were used to qualify crimes depending on the social position of the criminal), factual observations (i.e. women were running taverns).

Once encoded, our corpus can be considered as a semi-structured database of historical facts. Research was done in two steps: transcription and low-level encoding for building the database and analytical encoding and exploration for Social History research (for a longer description of the project in French, see [10].)

To create the database, the transcribed letters have been encoded using the XML/TEI format that offers a high expressive power for describing paratextual phenomena such as abbreviations, insertions, corrections and unreadable portions of text.

TEI is a set of rules, conventions and procedures for encoding textual documents for interchange among different fields of the humanities. It relies on SGML [12] and has been designed to gather and formalize the practices of various fields to avoid the proliferation of incompatible formats. TEI is now turning to XML with few changes for users thanks to the XML committee for ensuring close compatibility and to automatic document converters for performing the actual work.

Once encoded in XML/TEI, the beginning of the letter in Figure 1 looks like this:

```
<lb n="1"/><name reg=”Francois Ier”>Francoys
</name>, par la grace de Dieu, roy de France et
duc de
<lb n="2"/>Bretaigne, savoir faisons a tous
<abbr>presens</abbr> et a venir, nous avoir
<lb n="3"/>receu l'umble supplicacion et requeste
nous faicte des <s ana="structuresociale-parente">
parens et amys
<lb n="4"/>consanguins</s> de <abbr>notre</abbr>
povre subgect <name key="PL" ana="suppliant
masculin">Pierre Leserclier</name>, <s ana="crime-
statut-social">homme de labeur</s>
<lb n="5"/>de la paroisse d'<rs type="toponyme">
Oudon</rs> ou diocese de <rs type="toponyme">
Nantes</rs>, <abbr>contenant</abbr> que
<lb n="6"/>le <date value="30/9/1520" ana="crime-
date">dimanche, dernier jour de septembre dernier
</date>, <abbr>auquel</abbr> jour fut joué …
```

The encoding expresses three levels of phenomena: typographic or lexical, semantic and analytic.

1. Lexical and typographic phenomena: lines and pages are numbered using **lb** and **pb** tags, errors or omissions are noted (**sic**), as well as deletions (**del**), insertions (**ins**), unclear parts (**unclear**) and abbreviations (**abbr**);
2. Semantic encoding: names, dates and places are tagged as such and normalized using XML attributes. At this stage, the corpus is a database ready for interchange.
3. Analytical encoding: topics of interests are chosen by the historian and described in a sub-document. All segments of text in the corpus related to these topics are then linked to the topic of interest using the **ana** attribute. In the example above, the date of line 6 is marked as the date of the crime using the **ana** attribute.

This step requires the Historian to select a set of topics, describe them in a separate file. For example, the analytical topic "**crime-date**" is described in the sub-document by the following line:

```
<interp id="crime-date" type="date" value="Date of
the crime">
```

Topics are usually grouped using **interpGrp** (interpretation group) tags. For example, the "masculine" and "feminine" topics are grouped as "sex" like this:

```
<interpGrp id=”sex”>
 <interp id=”masculine” value=”male person”>
 <interp id=”feminine” value=”female person”>
</interpGrp>
```

Each portion of text related to each topic is then linked to the topic using the **ana** tag attribute, as in line 4 of the manuscript where the person is linked to both **masculin** and **suppliant**. When the region is already lexically or semantically tagged, like **name** in the previous example, the **ana** attribute is added in the element. Otherwise, an "**s**" element is created to delimit the region of interest as in line 4 and the **ana** attribute is set there. These mechanisms and their rationales are described at length in [23].

These encoding rules may seem very tedious but they are closely similar to the traditional practices in history, adding formal structure to the process of describing and analyzing manuscripts. Analytic encoding is a mechanism similar to maintaining manual files and notes for each document of the corpus. The traditional historical practice is not changed but adapted for the digital medium.

From there, traditional analysis of historic corpora is a well-established process whereby a historian starts from the sources and, through several levels of analysis, tries to find regularities or discrepancies and to structure the human history or find exceptions to supposed structures.

For this higher-level analysis, the historian is on her own and relies on her memories, files and notes. The Compus system has been designed to help and support the historian in this task of search and discovery.

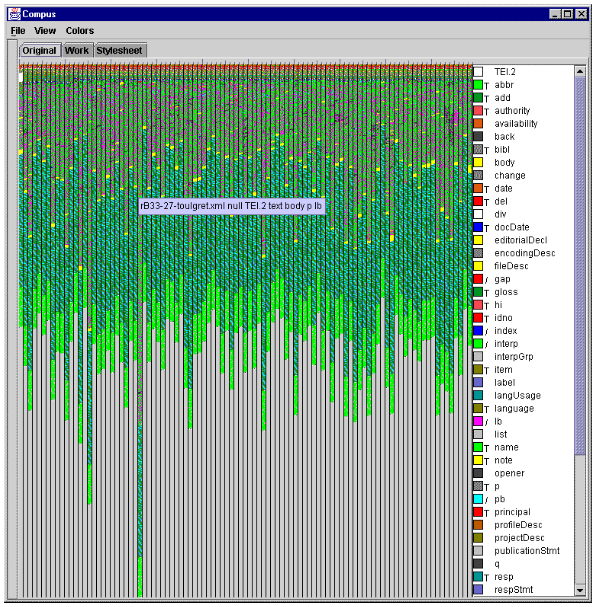


Figure 2: The Compus System showing a corpus of 100 structured documents. Four color patterns are visible for the TEI header, the text body, the glossary and the interpretation groups.

THE COMPUS SYSTEM

Figure 2 shows the Compus system. At startup, it contains three panes on a tab group. The main pane displays one vertical colored bar per XML document. Each XML element is assigned a color and a starting and ending index, corresponding to the character offset where the element starts and ends. A document is visualized as an ordered collection of colored segments using a space filling approach comparable to [15]. Since XML elements are nested, inner elements overlap outer ones.

For example, the sample XML document in boldface:

```
0    1    2    3    4
012345678901234567890123456789012345678901234567
<A>abcd<B>efgh</B><C>ijkl<D>mnop</D></C>qrst</A>
```

is first converted into the following list of intervals:
A=[0,48[, B=[7,18[, C=[18,40[, D=[25,36[
A color is then associated with each element name and the document is displayed. Each document is given the same amount of space, usually a thin vertical rectangle. The rectangle is considered as a long line that wraps. In this case, the line would have the following colors:
A: [0,7[, B: [7,18[, C: [18,25[, D: [25,36[, C: [36,40[, A: [40,48[

Wrapping the line every 5 pixels produces the following display:

Index	Color				
0	A	A	A	A	A
5	A	A	B	B	B
10	B	B	B	B	B
15	B	B	B	C	C
20	C	C	C	C	C
25	D	D	D	D	D
30	D	D	D	D	D
35	D	C	C	C	C
40	A	A	A	A	A
45	A	A	A		

For displaying a corpus, each document is displayed in a given order (date in our corpus) and is given the same rectangle. A scale factor is applied to the original indexes so that the longest document fits in the whole rectangle and all documents are comparable in length. Corpora displayable by Compus are limited by the screen size minus the width of the list box on the right, the scrollbar and a separator line, leaving room for about 500 documents on 1280x1024 screens. Our corpus contains 100 documents so each rectangle is 5 pixels wide, as shown in figure 3. More can be visualized either by scrolling –hiding some documents – or by splitting the screen vertically - thus trading more documents for less space per document.

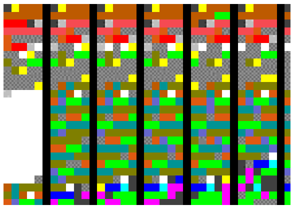


Figure 3: Close up view of the NW corner of figure 2.

By applying this visualization to a corpus, users can at once compare document sizes and overall structure. As Figure 2 clearly shows, four parts are visible on our corpus, due to a change in texture. They correspond to the TEI header, the body of the text, a glossary and the description of the analytical categories. The 27th letter is much longer than the others and the first letter exhibits a gray zone at the end of the header or at the beginning of the body. The length of the 27th letter has even surprised the historian who encoded it. A judge has been found guilty of prevarication and asks for a special grace (something very specific to 1531). The particular gray zone is a **revisionDesc** element containing the description of changes made to the document. The first document has been subject to more modifications than the others to fix rules we found suitable for the whole corpus. The mixed color of each part comes from the dominant encoding. The text part is full of abbreviations in maroon and dates in dark red. With just this first overview visualization, we can glean important clues about the contents of the corpus.

In the next subsections, we describe color allocation, how to interact with this representation, how to perform structural transformations and how to sort and export data, along with examples of uses.

Color Allocation

Each element type (i.e., tag name) is automatically assigned a color. Structured documents usually contain elements of various lengths, ranging from the full document to zero for empty elements. We compute the average size of each element and use three different color sets for large elements, middle sized elements and small elements. Large elements (over 100 characters long in average) use gray values, from black to white. Middle sized elements (from 10 to 100 characters) use the color scheme described by Healey [13]. Small elements use saturated red, green and blue colors. Threshold for color sets can be configured, but we found the 10/100 values well-suited to actual documents.

With this allocation scheme, larger elements are less visible than middle sized elements and small elements are vivid and can be distinguished. When more colors are required, the automatic color allocation cycles through the color sets. Interaction, as described in the next section, is useful to limit the number of visualized elements to improve search efficiency and focus on specific phenomena.

Interaction

ToolTips identify the document and the structural nesting of the pointed position. Clicking on the position displays the selected document in the pane called “Work”.

Color control and selection

Healey [13] has shown that 7 colors can be searched very quickly thanks to properties of our perceptual system. When a user starts Compus, her screen visualizes all the documents with more than 7 colors. Large-scale structures are visible through dominant colors, but users are usually searching for more precise phenomenon such as correlation or distribution of events. Moreover, when a color is assigned to each element name, nested elements disappear. For example, in `<name><unclear>Axe</unclear></name>`, the color of the **name** element will disappear under the color of the **unclear** element.

To control which elements are colored, the right list box displays the name of all the elements used in the corpus with their type and assigned color. Selecting an item hides the elements of that name on the visualization panel, revealing any element hidden behind it. Several items can be selected at the same time to focus on a smaller set of elements. Furthermore, users can control how colors are assigned to elements, either by right-clicking on the element in the list to popup a color selector or by triggering the color optimization menu that re-applies the color allocation to the currently visible set of elements.

With the list, users start by hiding unwanted elements and optimize colors to highlight specific phenomena. For example, selecting all elements but **unclear** in the scrollable list reveals documents harder to read because their density of colored segments is higher than the others. It turns out that our documents are all less readable at their end than at their beginning, due to the fact that they are handwritten copies of the original letters sent directly from the King to the requester and that the copyist was probably tired at the end. This distribution is visible from the variation of density of color on each document rectangle. By reducing the number of visible up to 7, users can improve their search time by relying on their perception system.

Visualization types

Because the hierarchical structure of elements creates a highly fragmented representation – i.e. a distribution – users might chose to see an aggregated view of the elements in the documents. With this visualization type, the surface covered by each selected element is visualized with bar graphs. In the example of figure 4 the corpus was filtered to show only the lexical elements (unclear, sic, deletions and additions), aggregated to reveal their relative importance in each document.

Sorting

Documents can be sorted according to a variety of attribute orders: e.g., number of visible elements, surface of visible elements and lexicographically according to visible elements. Number of elements is self-explanatory; surface is simply the sum of all visible elements sizes. Sorting by surface is useful for the **unclear** tag and shows documents from the hardest to read to the easiest. Lexicographic sort is more meaningful for transformed documents so we address it after the next section.

Once sorted, documents remain in that order until another sort is asked for. New regularities can be searched, like displaying a color for each copyist when the corpus is sorted

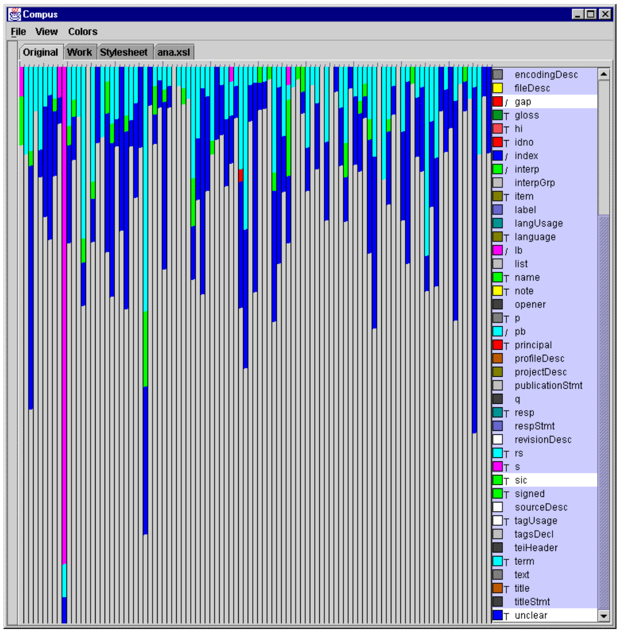


Figure 4: Aggregated elements view of lexical problems (add, del, sic, gap, unclear)

in readability order. Poor copyists appear on the left and good ones on the right.

Structural Transformations

At this stage, very few things can be said about social history because TEI elements describe lexical and syntactic phenomena. To access the analytic markup, attribute values are needed.

Compus allows for structural transformations of the documents to focus on specific elements, transform XML attributes into elements or perform more sophisticated transformations allowed by the expressive power of the XSLT [6] transformation language.

XSLT has originally been designed for expressing Style Sheets to format an XML document for printing or online display. XSLT applies rules to a source XML document to transform it into another XML document. We have integrated an XSLT processor into Compus to filter and refine the visualization of corpora.

XSLT rules have two parts: a match part and an action part. Conceptually, the XSLT processor does a traversal of the source document, searches for the most specific match at each node and applies the related action. This action returns a new structure or nothing. It may continue the traversal on sub elements of the source to build its new structure.

For example, only to process the body of a TEI document, the XSLT rule would be:

```
<xsl:template rule="/">
 <xsl:apply-templates select="//body"/>
</xsl:template>
```

This simple transformation extracts only the sub-tree starting at the **body** element of the document.

XSLT has two implicit rules saying that when no specific rule is defined, the text is copied and elements are ignored.

Since Compus only visualizes elements and not attributes, XSLT is used to change the structure of documents to transform events of interest into elements. Some elements can also be discarded or added to adapt the visualization. For each set of interrelated phenomena we want to visualize, we build an XSLT rule set that selects the useful elements and translates some attributes into elements that will be displayed. When focusing on analytical matters, we translate any TEI analytical attributes to elements of the same name. A document source containing:

```
<name ana="criminal">Jehan Mace</name>
```

becomes

```
<criminal>Jehan Mace</criminal>
```

by applying the following rule:

```
<xsl:template rule="*[@ana]">
 <xsl:element name="{@ana}">
 <xsl:apply-templates/>
 </xsl:element>
</xsl:template>
```

Where the rule reads: any element name with the **ana** attribute defined produces an element named by the contents of the attribute. The real rule is more complicated since **ana** attributes may contain a list of analytical names. We convert the list into nested elements or duplicated elements depending on the kind of visualization type. To visualize analytical phenomena, we also apply the following transformations: dates are normalized by replacing their contents by the contents of the **value** attribute. This regularization is required because the syntax of dates used in 1531 are hard to parse automatically (see line 6) and the calendar has changed from Julian to Gregorian. In a similar fashion, the transformation use regularized versions of the text when it exists, encoded in the **reg** attribute. Finally, we use the **n** attribute of elements analytically linked to ages to have plain numbers in the transformed version of the document.

New XSLT rules can be typed directly into Compus in a pane called “Stylesheet” or loaded from a regular text file. Since XSLT is a language, it is able to import modules so rule sets are usually small, about 30 lines long. When Compus applies the rule set to a list of source documents, it adds a new visualization pane similar to the original one, as shown in figure 5. Each pane has the name of its rule set so users may have more than one transformed pane if desired.

Once the rule set to visualize analytically the corpus is applied, we may want to focus on the life of women in the 16th century. We select only male and female elements in the element list, revealing the distribution of each. By selecting both criminal and feminine, we can see how many women are criminals (only two). The same can be done for victims (only two also). Relying on the corpus, crimes were a masculine matter.

Compus also infers the type of elements from their contents. Currently, Compus recognizes numeric types and dates. It also keeps track of empty elements, textual only elements and mixed elements. These types are displayed on the scrollable list next to the color square, using one letter per type: I for integer, R for real, D for date, T for text, / for empty elements and a space for mixed contents.

From there, users can search for new phenomena or export the new documents into files readable by spreadsheets or traditional information visualization tools such as Spotfire [2].

Sorting and Exporting

Compus can use element types inferred during structural transformation for two purposes: sorting and exporting.

When sorting with only one element visible, Compus uses the element type to perform the sort. When several elements are visible and all are typed, Compus asks for an order and sorts the documents using a lexicographic order.

The initial order of our corpus is by administrative date of decision, but the date of the crime is also marked using an analytical link. Therefore, we can sort crimes according to their type first and then to their date.

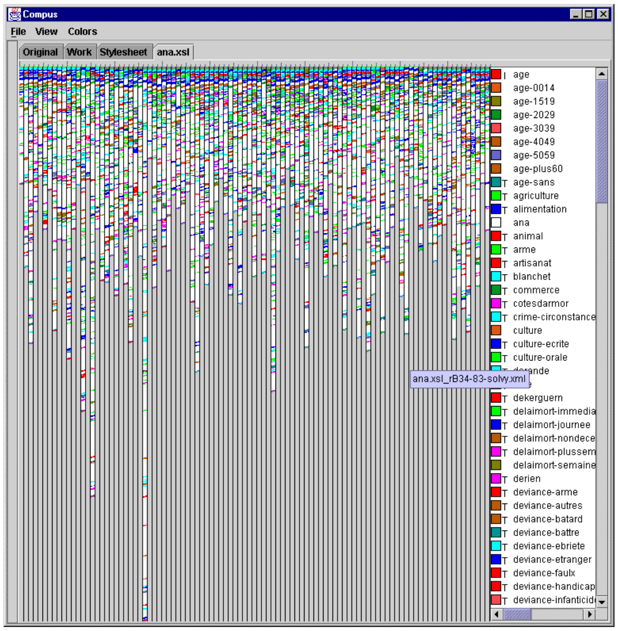


Figure 5: Visualization of the transformed corpus, i.e. analytical view of figure 2. Only the text body is visible. Colored element names come from analytical attributes not visible from figure 2.

When a sort is applied, all the views are sorted so one view can be used to express an order that may reveal a phenomenon visible in another view.

Element types are also used to export filtered documents. A traditional table is then produced containing one row per document and one column per selected element name. When a document contains several instances of a selected element, they are concatenated for text elements and an error is signaled for the other types.

We have exported views to perform calculations using a spreadsheet calculator and for other types of visualizations like showing the delay between the crime and the decision as shown in Figure 6. This delay decreased probably because the King wanted to keep Brittany happy just before it joined the French Kingdom. However, some crimes were old with a peak of 270 months when the criminal escaped, returned and got caught more than 22 years later.

Implementation and performance

Compus is implemented in about 5000 lines of Java 1.2 and relies on the XP Java library for parsing XML and the XT system for applying XSLT transforms[1]. It has been used on 300Mhz PCs with at least 128Mb of memory required to load the documents for fast redisplay during dynamic queries.

Applying the transformation of Figure 4 to the 100 documents takes about 15 seconds. Users never complained about performance, probably because the exploration and analysis are much longer.

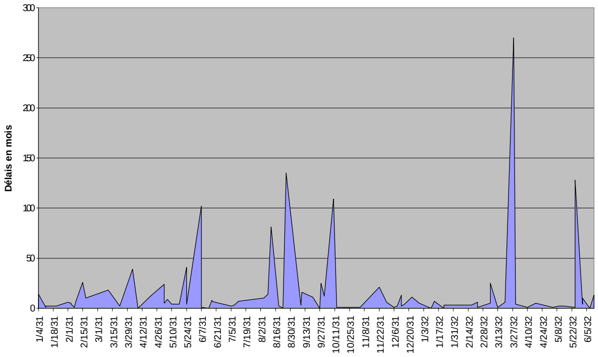


Figure 6: delay in months between the crime and its grace visualized from a Compus export.

RELATED WORK

Compus is related to Seesoft [9], Spotfire [2], the Document Lens [22] and the system described by Lecolinet et al. in [17].

Seesoft visualizes documents using vertical bars and colors. However, SeeSoft and Compus are quite different in their goals and details. Seesoft displays statistics associated with lines of code and extracted from an external database. Most problems with structured documents come from their dual aspects as databases and text. Compus uses a space filling approach for displaying a corpus whereas Seesoft clearly separates each document, providing a lower information density. Nested elements are not addressed by Seesoft, neither is color allocation and optimization.

Spotfire is a commercial system that implements the concept of starfield display with dynamic queries. It displays a database table using several types of visualizations (scatter plots, pie charts, histograms and bar charts) and provides interactive controls to filter visualized items. Large databases can be explored through these visualizations. Dynamic filtering helps in finding regularities and discrepancies by visual inspection, relying on our perceptual system. Compus is a specialization of the concept of starfield display with dynamic queries: it displays a corpus of structured documents instead of a flat database table, using a particular type of visualization not provided by Spotfire. The dynamic selection of visible elements is available in Spotfire but not the color allocation scheme. Spotfire can extract views from a data base by SQL queries whereas Compus extracts views from an XML corpus by applying XSLT transforms.

[1] XP and XT are two free Java libraries and programs developed by James Clark and available at the following URL: http://www.jclark.com/xml/

The Xerox Document Lens is part of the WebBook environment and can display several pages using a perspective visualization. However, it is intended for displaying formatted Web pages and not structured documents: it is page oriented and not corpus oriented and does not contain functions to compare or sort documents since it relies on the natural page order.

Lecolinet et al. have designed a system to display a corpus for comparing text variants. A perspective wall is used to display each page of a document next to the other on a line and the documents stacked in columns. This representation simplifies the reading of aligned portions of text but is not adapted to corpus on unrelated documents like ours. Compus does not provide a focus+context view but a space filling view of the corpus. For interaction, Lecolinet's system relies on navigation in a static representation whereas Compus uses dynamic queries and structural transformations. The level of granularity is also different: Lecolinet focuses on aligned pages of a variant document whereas Compus focuses on XML elements contained in several documents.

DISCUSSION

Compus has been used by historians to initially perceive the relative importance of each phenomenon, then to quickly check hypothesis about correlation of phenomena or spot discrepancies, as described in the second section. Quickly checking for hypothesis is an important property of Compus since it is difficult for researchers to rely entirely on their memory to judge the frequency of events. Important events are often judged frequent and conversely frequent unimportant events are usually left out.

For example, our initial hypothesis was that just before 1532, more nobles would receive clemency (i.e., be pardoned) for political reasons. Compus helped reveal that the density of pardon given to nobles did not vary across the two years of the corpus. However, two important nobles did receive a pardon in the last letters. We have probably been influenced by these events to produce our hypothesis.

We will now describe the interaction of the various tools and some other experiments we have done on other corpora: the plays of Shakespeare and an Old English language corpus. Finally we will address some limitations of Compus.

Tools for working on structured documents

Researchers need to access several levels of granularity when working on a document corpus: the corpus, the document and various fragments of documents. Compus is suited to visualize a whole corpus but other tools are needed to read documents and their parts. One very effective view of the corpus is through a Web browser and indexes generated from the TEI documents. Figure 7 shows such a configuration.

We also mentioned spreadsheet calculators and information visualization systems to assist in the analysis of valued information, such as dates, age, time, frequency, etc.

Synoptic visualization has been found effective to explore a new corpus. This will become a more and more common situation with the development of digital libraries where users will need to quickly evaluate the scope of collections.

Exploring Shakespeare

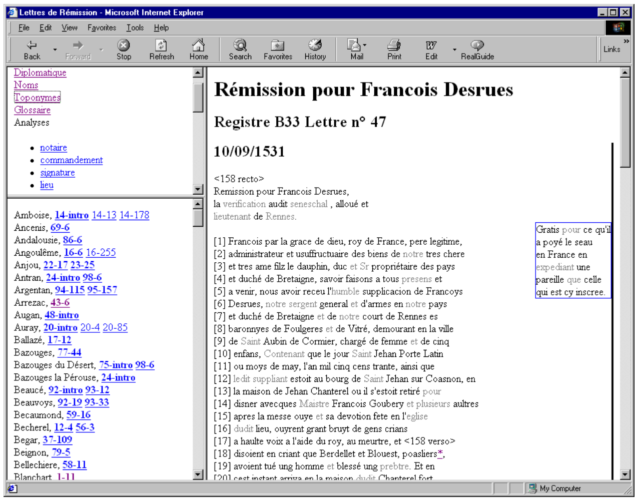
We tested Compus on a corpus of 37 plays of Shakespeare encoded in XML [3]. Variation in sizes is obvious. "Henry the Eighth" exhibits larger **STAGEDIR** (stage direction) elements than the others, meaning that scenes happen in different places or that lots of events happen during the scenes. Even if the level of encoding is much lower than ours, Compus is still useful.

The Lampeter Corpus

The Lampeter corpus [20] "is an unusual historical corpus, consisting of 120 unique English pamphlets from the period 1640 to 1740." We received it as an answer to a request on the Internet for testing Compus. Without any knowledge of the corpus, we have been able to see several classes of documents. The corpus is organized into 6 topics: religion, politics, economy and trade, science, law and miscellaneous. We found that some structures were topic specific, like tables used mostly for science, but also for economy. Since the corpus has been selected for its use of the language, we applied a transformation to exhibit language use per topic. Figure 8 shows its visual form when applied a suited analytical filter. This visualization reveals that Latin was by far the most used language (not a surprise). That Greek was mostly used for religion and science whereas Latin was used everywhere but mostly for science, religion and law. French language was most used in political texts, revealing the tensions.

Limitations

One of the current limitations of Compus is that the transformation language XSLT requires either extensive learning or technical assistance. It will not remain such a problem if technologies related to XML reach a level of popularity comparable to HTML but providing a simpler



Diplomatique
Noms
Toponymes
Glossaire
Analyses

- notaire
- commandement
- signature
- lieu

Amboise, 14-intro 14-13 14-178
Ancenis, 69-6
Andalousie, 86-6
Angoulême, 16-6 16-255
Anjou, 22-17 23-25
Antran, 24-intro 98-6
Argentan, 94-115 95-157
Arrezac, 43-6
Augan, 48-intro
Auray, 20-intro 20-4 20-85
Ballazé, 17-12
Bazouges, 77-44
Bazouges du Désert, 75-intro 98-6
Bazouges la Pérouse, 24-intro
Beaucé, 92-intro 93-12
Beauvoys, 92-19 93-33
Becaumond, 59-16
Becherel, 12-4 56-3
Begar, 37-109
Beignon, 79-5
Bellechiere, 58-11
Blanchart, 1-11

Rémission pour Francois Desrues

Registre B33 Lettre n° 47

10/09/1531

<158 recto>
Remission pour Francois Desrues,
la verification audit seneschal , alloué et
lieutenant de Rennes.

[1] Francois par la grace de dieu, roy de France, pere legitime,
[2] administrateur et usuffructuaire des biens de notre tres chere
[3] et tres ame filz le dauphin, duc et Sr propriétaire des pays
[4] et duché de Bretaigne, savoir faisons a tous presens et
[5] a venir, nous avoir receu l'humble supplicacion de Francoys
[6] Desrues, notre sergent general et d'armes en notre pays
[7] et duché de Bretaigne et de notre court de Rennes es
[8] baronnyes de Foulgeres et de Vitré, demourant en la ville
[9] de Saint Aubin de Cormier, chargé de femme et de cinq
[10] enfans, Contenant que le jour Saint Jehan Porte Latin
[11] ou moys de may, l'an mil cinq cens trante, ainsi que
[12] ledit suppliant estoit au bourg de Saint Jehan sur Coasnon, en
[13] la maison de Jehan Chanterel ou il s'estoit retiré pour
[14] disner avecques Maistre Francois Goubery et plusieurs aultres
[15] apres la messe ouye et sa devotion fete en l'eglise
[16] dudit lieu, ouyrent grant bruyt de gens crians
[17] a haulte voix a l'aide du roy, au meurtre, et <158 verso>
[18] disoient en criant que Berdellet et Blouest, poasliers*,
[19] avoient tué ung homme et blessé ung prebtre. Et en
[20] cest instant arriva en la maison dudit Chanterel fort

Gratis pour ce qu'il
a poyé le seau
en France en
expediant une
pareille que celle
qui est cy inscree.

Figure 7: Hypertext configuration with various indexes to read and explore the corpus.

interface to perform the transformations would certainly help users.

Color allocation should be improved. We observed that users often need to allocate colors using the popup chooser. More investigations are required here.

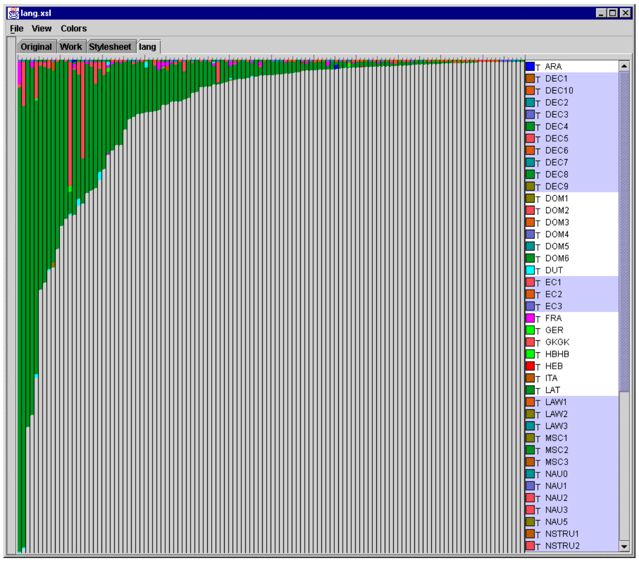


Figure 8: The Lampeter Corpus visualized by element surface showing the various amounts of non-English languages used in descending order. Latin is by far the most popular followed by Greek.

It would be useful to manage larger corpora from Compus, although most homogeneous corpora analyzed by current historians are usually under 1000 documents, due to the time required to fill analytical forms manually. As explained in section 2, Visualizing more than 1000 documents could be done either by scrolling or splitting the screen vertically. We are interested in investigating both solutions but have not yet found a homogeneous corpus analytically encoded larger than 1000 documents to experiment with.

CONCLUSION AND FUTURE WORK

We have described Compus, a new visualization system suited to exploring corpora of structured documents. It has been mostly experimented on a corpus of historic documents but has also been tested on Shakespeare's plays and a literary corpus.

Compus displays an XML corpus by assigning colors to each element name and applying a space filling representation that visualizes the position and size of each element in each document. Interaction is used to focus on elements of interest. Since only XML elements are visualized, Compus integrates an XSLT processor that transforms structurally XML documents in other XML documents. Phenomena expressed through several encoding mechanisms can be translated into elements and visualized. Transformations are also useful to filter documents content and focus on specific parts of documents or on a set of tags. From there, users can notice global correlation and discrepancies. Compus can also sort document representations according to several orders to reveal other types of correlations.

We have shown how Compus has been successfully used by researchers in early modern history to explore a corpus made of 100 documents encoded in XML/TEI. We have also shown how Compus could be used as a visual interface for accessing databases of homogeneous structured documents. We are now exploring a new application domain with a corpus on biology and will study how practitioners use Compus.

Several library projects are working on such corpora or databases and their access is currently through forms and lists, comparable to FTP access before the Web existed (except for [18]). Compus is an effective tool for exploring the specific class of homogeneous corpora where it provides a global view and several insights to the contents.

We believe Compus is an effective complement to already existing tools like indexers, style sheet processors, spreadsheet calculators and information visualization programs. Rather than building an integrated environment to work on structured document corpora, Compus can be used as a visual explorer, and delegate specific tasks to the other tools.

As for Social life in the 16th century in Brittany described by the Letters of Clemency, alcohol was the main cause of murders. Women were often managing the houses, sometimes even taverns where both men and women were drinking. Comments are left to the patient reader.

ACKNOWLEGMENTS

Thanks to Catherine Plaisant, Wendy Mackay and Cédric Dumas for their advice and help in writing the article. Thanks to Stephanie Evans for cross proofing it.

REFERENCES

1. Christopher Ahlberg and Ben Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In Human Factors in Computing Systems. Conference Proceedings CHI'94, pages 313--317, 1994. ACM.
2. Christopher Ahlberg Spotfire: An Information Exploration Environment SIGMOD REPORT, 25(4), pp. 25-29, December 1996.
3. Bosak J. Shakespeare 2.00. Available at http://metalab.unc.edu/bosak/xml/eg/shaks200.zip
4. Bray, T. Paoli, J. Sperberg-McQueen C. M. Eds. Extensible Markup Language (XML) 1.0, Recommendation of the W3 Consortium. Feb. 1998. Available at http://www.w3.org/TR/REC-xml
5. Caton, Paul. "Putting Renaissance Women Online," New Models and Opportunities, ICCC/IFIP Working Conference on Electronic Publishing '97, April 1997. See also at http://www.wwp.brown.edu/

6. Clark, J. XSL Transformations (XSLT) Version 1.0 W3C Working Draft.
Available at http://www.w3.org/TR/WD-xslt

7. Cover, R. The SGML/XML Web Page, Available at http://www.oasis-open.org/cover/.

8. Dufournaud, N. Comportements et relations sociales en Bretagne vers 1530, d'après les lettres de rémission, Mémoire de Maitrise, Univ. De Nantes. Available at http://palissy.humana.univ-nantes.fr/cete/txt/remission/Memoire.pdf

9. Stephen G. Eick and Joseph L. Steffen and Eric E. Sumner Jr. Seesoft --- A Tool For Visualizing Line Oriented Software Statistics IEEE Transactions on Software Engineering, pp. 957-68, November 1992.

10. Fekete, J.-D. and Dufournaud N. Analyse historique de sources manuscrites : application de TEI à un corpus de lettres de rémission du XVIième siècle Special issue "Les documents anciens", (Hermès), vol.3, 1-2, 1999, pp. 117-134 (in French).

11. Friedland, L. E. and Price-Wilkin J. TEI and XML in Digital Libraries, Workshop July 1998, Library of Congress. Available at http://www.hti.umich.edu/misc/ssp/workshops/teidlf/

12. Charles F. Goldfarb and Yuri Rubinsky The SGML handbook, Clarendon Press, 1990.

13. Christopher G. Healey Choosing Effective Colours for Data Visualization Proceedings of the Conference on Visualization, pp. 263-270, IEEE, October 27- Nov 1 1996.

14. Nancy Ide and Dan Greenstein, Eds. Tenth Anniversary of the Text Encoding Initiative, Computer and the Humanities, 33(1-2), 1999.

15. Keim D. A.: Pixel-oriented Database Visualizations , Sigmod Record, Special Issue on Information Visualization, Dec. 1996.

16. Ian Lancashire, John Bradley, Willard McCarty, Michael Stairs, Using TACT with Electronic Texts. New York: MLA, December 1996.

17. E. Lecolinet and L. Likforman-Sulem and L. Robert and F. Role and J-L. Lebrave An Integrated Reading and Editing Environment for Scholarly Research on Literary Works and their Handwritten Sources DL'98: Proceedings of the 3rd ACM International Conference on Digital Libraries, pp. 144-151, 1998.

18. Marchionini, G., Plaisant, C., Komlodi, A. Interfaces and Tools for the Library of Congress National Digital Library Program Information Processing & Management, 34, 5, pp. 535-555, 1998.

19. The Oxford Text Archives, available at http://ota.ahds.ac.uk/.

20. Siemund, Rainer, and Claudia Claridge. 1997. "The Lampeter Corpus of Early Modern English Tracts." ICAME Journal 21, 61-70., Norwegian Computing Centre for the Humanities.

21. Randall M. Rohrer and David S. Ebert and John L. Sibert The Shape Of Shakespeare: Visualizing Text Using Implicit Surfaces Proceedings IEEE Symposium on Information Visualization 1998, pp. 121-129, 1998.

22. George G. Robertson and Jock D. Mackinlay The Document Lens Proceedings of the ACM Symposium on User Interface Software and Technology, Visualizing Information, pp. 101-108, 1993.

23. C. M. Sperberg-McQueen and Lou Burnard (eds.) Guidelines for Electronic Text Encoding and Interchange (TEI P3), Volumes 1 and 2, The Association for Computers and the Humanities, the Association for Computational Linguistics, and the Association for Literary and Linguistic Computing, 1994.