Text-Image Coupling for Editing Literary Sources

Eric Lecolinet, Laurent Robert
Ecole Nationale Supérieure
des Télécommunications
& CNRS URA 820
46 rue Barrault, 75013 Paris, France
elc,lrobert@enst.fr

François Role
Université Paris X
& INRIA (ATOLL project)
Francois.Role@inria.fr

Abstract

Users need more sophisticated tools to handle the growing number of image-based documents available in databases. In this paper, we present a system devoted to the editing and browsing of complex literary hypermedia including original manuscript documents and other handwritten sources. Editing capabilities allow the user to transcribe manuscript images in an interactive way and to encode the resulting textual representation by means of a logical markup language (based on the XML/TEI specification). Both representations (image and structured text) are tightly linked to facilitate the reading and the interpretation of documents. This text/image coupling scheme is an attempt to unify several layers of information in order to provide the user with a global vision of the work. Our system also supplies tools capable of processing and relating information stored both in images and structured texts. Finally, application-specific visualization techniques have been developed in order to provide users with a way to identify relationships between source documents and help them to navigate.

Keywords

Image-based humanities computing, transcription/editing tools, text/image coupling, text encoding and rendering, XML, TEI, browsing and navigation in large hypermedia.

1. INTRODUCTION

1.1 Context

Hypermedia technology now makes it possible to create electronic documents made from a variety of heterogeneous sources such as texts, images, sounds, etc. This represents a great opportunity for the development of hypermedia archives of literary sources, as shown by the increasing number of Digital Libraries projects providing access to high-quality images of primary source materials. For example, the Canterbury Tales [2], the William Blake Archive [3] and the Electronic Beowulf [7] projects are dealing with large databases of digital images and related documents (images or printed texts), to mention just a few. These hypermedia archives allow for the representation of primary literary works that are disparate, dispersed and difficult to access.

However, while setting up and organizing digital repositories to provide better access to vast amounts of heterogeneous data is a major topic in digital library research, it seems that less work has been devoted to the development of integrated environments for scholarly research on this material. This article describes an environment for editing, visualizing, and integrating the multiple sources that compose modern literary hypermedia. This prototype was initially developed as part of the Philectre project [11,25] whose goal was to explore computer science techniques for scholars. Hence, this tool is primarily intended for the editors of hypermedia works and more specifically for scholars performing literary research. But such tools could also be used by a broader audience as they bring new capabilities to hypermedia readers. These capabilities, which facilitate the enrichment and the appropriation of the initial document by the reader, are also related to hypertext annotation [16] and to active reading, another promising research field [21].

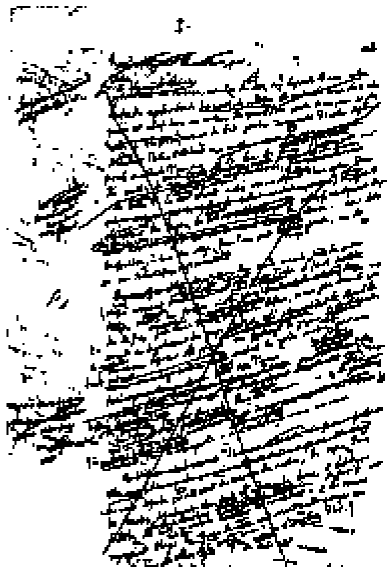


Figure 1: A page from a Flaubert's manuscript.

The presented work focuses on the editing of image-oriented hypermedia that contain many document images (such as original manuscript facsimiles) and require efficient means for relating these images with textual parts. This is related to the dual nature of manuscripts as they can either be seen as "pure" textual representations or as bitmap images (that actually contain text and graphical elements that can be interpreted by readers). For instance, as shown in Figure 1, manuscript images can contain blocks of text that are not arranged in a linear way. Notes, diagrams, images, corrections or references can be located in the margin or between a couple of lines. Various blocks of text can possibly be linked together with arrows or other graphical markings, thus making reference to each other. Certain words, sentences, paragraphs may have been crossed out or damaged and various components can have been added by various people. Hence, manuscripts are *texts with a graphical interest* [1] that require specific tools and techniques for taking into account this dual (i.e. textual and graphical) nature and for providing ubiquitous to access both types of representations.

The handwritten sources of literary works that can be found in manuscripts are complex objects that may quite hard to decipher. This difficulty is not only due to the graphical complexity of manuscripts but is also related to the multiplicity and the variety of sources and the relationships between these documents and other data. For instance, authors often produced several preliminary *versions* of the same page. Similarly, several *variants* of ancient texts are often available. And these manuscripts are generally related to a number of other sources. Thus, for literature specialists or philologists, literary works can not be reduced to "canonical texts" but consist in large sets of heterogeneous data (including the handwritten copies of ancient works, the authorial manuscripts of modern works, transcriptions, translations, scholarly editions, commentaries, etc.). These specific needs clearly show the importance of the underlying encoding scheme that is used for storing and organizing the data. Paradoxically, this encoding scheme must be as general and standardized as possible (in order to improve tool compatibility, information sharing and cooperative work) but must also provide powerful (and thus specific) means for tightly coupling the manuscript images with their textual counterparts. Besides, text/image coupling raises theoretical questions as to the traditional distinction between physical and logical structures.

Lastly, visualizing and exploring large information spaces are difficult tasks. The complexity of literary hypermedia, and the multiplicity and variety of sources they generally contain may cause disorientation problems to the reader. This problem must be taken into account by providing advanced tools that facilitate the browsing and the navigation in such complex information spaces.

The following subsections describes related work, the design principles of the proposed system and the organization of the article.

1.2 Previous Work

The "PLAO" system developed within the context of the new French National Library project in Paris between 1990 and 1993 can be recalled as one of the few projects that dealt explicitly with this issue [26]. Although this project was not brought to full completion it inspired other systems specifically intended to support digital scholarship.

The BAMBI system [5,6] (Better Access to Manuscripts and Browsing of Images) was developed within the European Community Research Program on Digital Libraries from 1994 to 1997. It is intended to support work on ancient texts available both in text and image format. More precisely, it is a software that allows philologists and codicologists to transcribe and annotate classical and medieval manuscripts. BAMBI provides the user with a transcription tool and a document analysis module which allows for the automatic segmentation into lines (and words in the case of manuscripts with a regular and neat layout). By using the transcription and the results of the document analysis module, a word-image concordance can be produced in a semi-automatic way. This makes it possible to navigate between the words of the transcription and the matching pieces in the image. The system also makes it possible to attach annotations (comments, critical apparatus) to the transcription.

Another editing environment called MVED (Multi-variant Contents Editor) is presented in [9]. As its name implies, MVED has been designed to support the editing of documents with multiple variants. This editing tool has been used to build an electronic edition of Don Quixote that is based on old books that were published during Cervantes' lifetime. It includes a collator module, for comparing textual variances among multiple versions as well as a batch text and image synchronization mechanism at the page and line levels. In addition, it provides capabilities for editing and classifying variants and for recording commentaries about the editing transactions.

Debora [4] and Inote [10] are another example of tools that allow the scholarly user to attach textual annotations to various regions in an image and then store those annotations in electronic form.

1.3 Design Principles and Plan of the Paper

Among the most successful encoding schemes are logical markup languages which make it possible to define common languages for a community of researchers, and thus facilitate common work on this data. Besides, the use of markup languages can also improve automatic processing when using search engines and other programs for building index tables, for computing statistics and so on. However, such languages generally provide limited ways of representing the graphical characteristics of the source text. This limitation has the following causes: first, markup languages were initially designed for representing printed text with a well-understood structure (for instance the usual book / chapter / section / paragraph structure) that is basically one-dimensional (in contrast to manuscript images where text, annotations, remarks, corrections are freely laid out in a two-dimensional way.) Second, encoding all the graphical and spatial characteristics of the manuscript image with a markup language would be quite a tedious and ambiguous task prone to many errors and misinterpretations.

These problems are solved in the following way: besides classical markup and hypermedia capabilities, the presented system implements techniques for coupling the structured text with the corresponding image subparts. This can be seen as an attempt to unify internal (encoded) and external (image) text representations by providing a two-way mapping between these representations. In that sense, this scheme facilitates "access by content" to the image, as image subparts can be retrieved through their linking to the text encoding. So, it is possible to process text by using traditional software while retaining all the visual information. This integration is needed in order to be able to interpret literary sources in a comprehensive way.

The text/image coupling scheme is based on two complementary procedures. First, human-computer interaction and document analysis techniques are provided for determining the visual correspondence between the graphical and the textual elements. This process, and the tools that were developed for performing this task, will be described in section 2. During this presentation, we will focus on the transcription editing task of literary material. Second, the XML/TEI markup language is used for encoding the text and for storing the corresponding text/image coupling data. This encoding scheme will be detailed in section 3. The problems related to image zones addressing will be highlighted, and we will explain how we circumvented them by augmenting the TEI pointer semantics. We will also present the advantages of using a well-known markup language that has been widely used for the humanities such as XML/TEI. Indeed, this choice of design favors the sharing of data among various projects and makes it possible to use a variety of XML-compatible complementary tools.

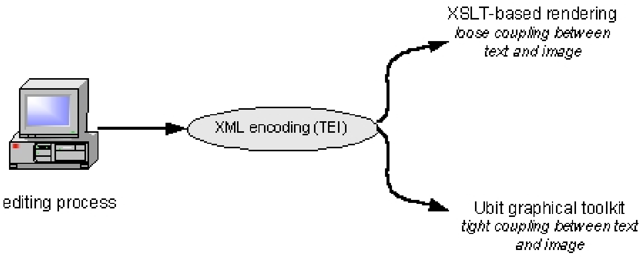


Figure 2: System overview.

This design also facilitates the distribution of edited materials, especially for developing Web based solutions as XML browsers or translators are now commonly available. Rendering aspects, by using either simple Web-based browsing tools or more sophisticated (and specific) visualization software (Figure 2) will be addressed in section 4. This section will first show how XSL transformations can be used for this purpose. It will then present the capabilities of the Ubit toolkit (a new GUI toolkit we developed) for creating XML editing and browsing tools.

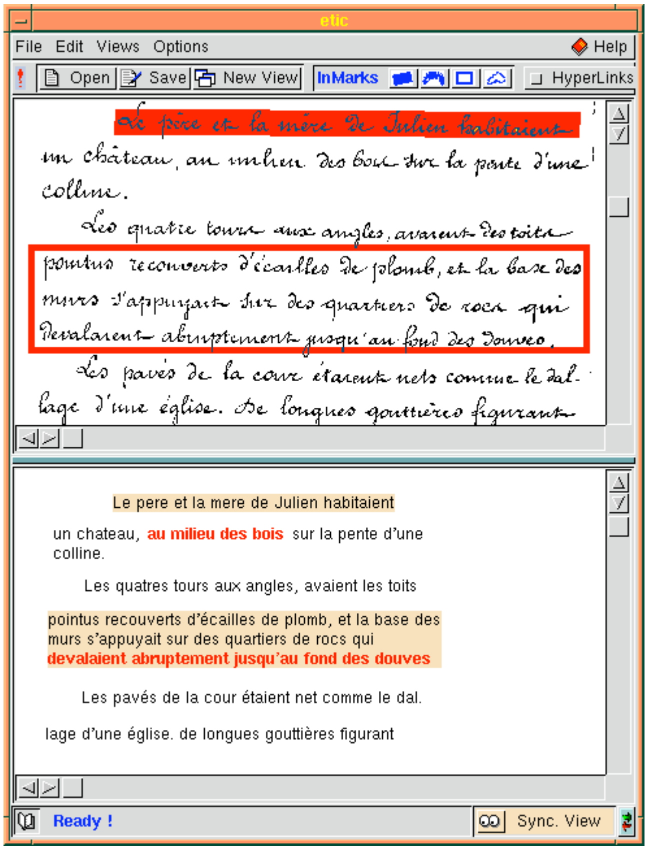
We will then propose new visualization techniques for navigating in large hypermedia spaces (section 5) and we will conclude.

2 INTERACTIVE EDITING

2.1 Text/Image Coupling

As mentioned before, manuscript images convey more information than their "pure" textual content. Their spatial layout and the graphical tracks they contain often provide pertinent information that can facilitate their interpretation. This underscores the difficulty of transcribing manuscripts efficiently with a classical editing process.

The transcription task consists in editing a typed version of a manuscript from its original image, while preserving its graphic aspect (the spatial disposition of text lines must be preserved as well as erasures and other editing marks). The transcription can be seen as an intermediate state between image and text: it is text with topological attributes.



Le père et la mère de Julien habitaient
un château, au milieu des bois sur la pente d'une
colline.
Les quatre tours aux angles, avaient des toits
pointus recouverts d'écailles de plomb, et la base des
murs s'appuyait sur des quartiers de rocs qui
devalaient abruptement jusqu'au fond des douves.
Les pavés de la cour étaient nets comme le dal-
lage d'une église. De longues gouttières figurant

Le pere et la mere de Julien habitaient

un chateau, au milieu des bois sur la pente d'une
colline.

Les quatres tours aux angles, avaient les toits

pointus recouverts d'écailles de plomb, et la base des
murs s'appuyait sur des quartiers de rocs qui
devalaient abruptement jusqu'au fond des douves

Les pavés de la cour étaient net comme le dal.

lage d'une église. de longues gouttières figurant

Figure 3: Text/image coupling in the transcription/editing tool. Links are both highlighted in the manuscript image (top half) and in the transcription (bottom half) when the user moves the mouse or clicks in the sensitive areas.

The transcription/editing tool makes use of human-computer interaction and hypermedia capabilities in order to simplify and improve this task. Hypermedia makes it possible to attach relevant information such as notes, comments and other related documents (containing text, images, sound) to the text that is being studied. Graphical user interface techniques provide efficient means for coupling the textual data and the manuscript images that contain the original representation of this text. In this model, the document components are interactively connected by adding graphic tracks (i.e. sensitive areas) in the manuscript image. These graphic tracks can be laid down in a direct manipulation style by using tools called interactive markers. These tracks can then be associated with other parts of the document, or with external data. They correspond to the anchors of the hyperlinks. Two types of links can be defined. The *transcription links* associate image elements in the manuscripts with their corresponding textual elements in the transcription (Figure 3). The *external links* provide a way to anchor heterogeneous hypermedia entities (texts, images, sounds, videos, Web pages).

As the transcription task basically consists in associating image areas with the corresponding transcribed text, it is quite important to provide efficient tools for adding graphics tracks in the manuscript image. Several interactive markers have been designed for selecting image zones in various ways. The first two markers can select straight or fluctuating text lines in manuscript images. They can also be used to select interesting words or word groups inside these lines. The first marker, which selects thin rectangular areas, is convenient for selecting regular lines (as shown in the top of Figure 4). The second marker, which follows the user's freehand drawing, makes it possible to designate fluctuating and irregular text lines. The two other interactive markers can be used for framing several lines of text or arbitrary areas in the manuscript image. The rectangular marker makes it easy to select line groups that are laid out in a regular way while the "lasso" marker (bottom of Figure 4) is useful for defining areas with arbitrary shapes.

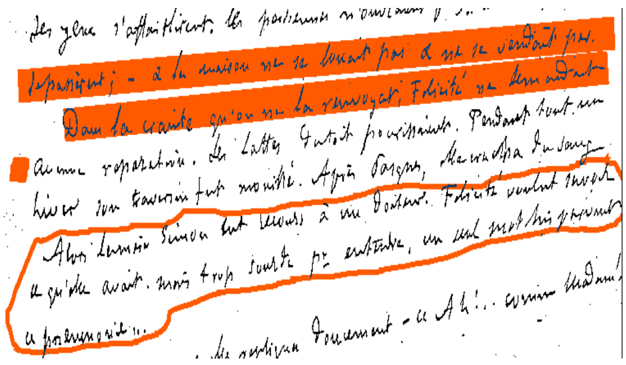
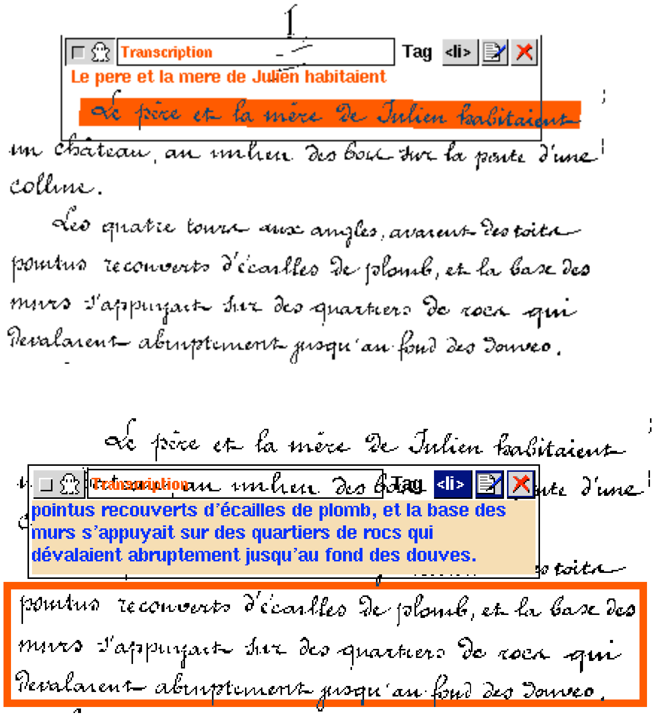


Figure 4: Interactive markers.

The transcription box provides the user with an interface for entering the corresponding text. This tool is automatically positioned near the selected area once a graphic track has been added. As illustrated in Figure 5, this tool can be rendered in transparent mode in order to increase the visual proximity effect between the manuscript image and the GUI area where the user is entering the text. The user then gets the impression that he is writing on the same document (as if he was annotating a "real" document). Besides, this rendering scheme avoids occlusion and lets contextual data remain visible while transcribing a given image zone. However, as transparency may be undesirable in certain cases (e.g. when the background pattern is too complex) this feature can be deactivated at any time.



Transcription

Le pere et la mere de Julien habitaient

Le père et la mère de Julien habitaient un château, au milieu des bois sur la pente d'une colline.

Les quatre tours aux angles, avaient des toits pointus recouverts d'écailles de plomb, et la base des murs s'appuyait sur des quartiers de rocs qui dévalaient abruptement jusqu'au fond des douves.

Transcription

pointus recouverts d'écailles de plomb, et la base des murs s'appuyait sur des quartiers de rocs qui dévalaient abruptement jusqu'au fond des douves.

Figure 5: The transparent transcription box (top half). Transparency can be deactivated when preferable (bottom half).

2.2 Linking and Automatic Layout capabilities

The system provides automatic layout capabilities in order to facilitate the creation of "diplomatic" transcriptions. The textual data that has been associated with image areas is automatically located in a way that respects the spatial disposition in the original manuscript image.

This feature is related to the text/image linking mechanism which will be detailed in the following sections. Two-directional links, which associate image areas with encoded text, are implicitly created during the transcription phase. As these links must necessarily store the image coordinates of the graphic tracks, this data can also be used for displaying the textual components in a way that approximates the layout of the manuscript image. It must be noticed however that this feature is optional as textual components can be displayed in various ways (such as ordinary "plain text" or fully "encoded text", as will be shown in section 4).

The user can take advantage of these coupling links during the editing and visualization phases. For instance, corresponding zones in the manuscript and textual views are automatically shown when selecting areas in any of these views. The way this correspondence is visually rendered depends on interactive settings. The *coupled-view mode* (Figure 3) makes it possible to display the manuscript image and the corresponding transcription in separate windows (or, alternatively, in separate boxes located in the same window). Sensitive areas are then implicitly detected by moving the mouse in any box and related areas are automatically highlighted in all representations. For instance, moving the mouse in the manuscript image will automatically highlight the graphic tracks that were previously defined in this image and the corresponding areas in its textual transcription (and vice versa). This mechanism is quite general as the system makes it possible to open multiple views of the same data. It is for instance possible to show simultaneously an overview of the manuscript image (by opening a reduced, low-resolution view), a zoomed view (that would display the same manuscript with a better resolution), the corresponding transcription, the XML encoding, etc. All views would then be implicitly linked and highlighted in a coherent way.

The system also provides other coupled visualization schemes. The *tool-tip mode* only shows the manuscript image. The transcription is not displayed in a second view, but the textual components that are linked to a specific graphic track are displayed as "tool tips" when the mouse enters corresponding areas in the manuscript image. The *superimposition mode* displays both the manuscript and the transcription in the same view by superimposing them. However, this mode may cause readability problems if the manuscript image is too complex or was written in a very irregular way (as in Flaubert's manuscript shown in Figure 1).

2.3 Semi-automatic approach

As dealing with large documents in a fully interactive way could become a tedious task, we have developed a semi-automatic approach that lets the user validate or correct interactively transcription hypotheses that are obtained from automatic document analysis. This capability facilitates interactive coupling by pre-segmenting manuscript images into potential line (or word) areas. As handwritten documents often have quite a complex structure, it is generally not possible to process them in a fully automatic way. As a consequence, user validation and correction is needed with most documents (and especially with modern handwritten manuscripts). Thus, semi-automatic approaches that give the final answer to the user (and let her activate or deactivate automatic tools) should be preferred to fully automatic approaches and the focus should be put on flexible integration schemes between automatic and interactive approaches.

Classical OCR and document analysis methods, which were mainly developed for printed material, can hardly be applied to handwritten documents. These methods are generally unable to process images that contain fluctuating or interwoven lines, irregular spacing between lines and words, insertions between lines or include several text line directions. Besides, most character recognition methods only work with well separated machine- or hand- printed characters. Cursive handwritten recognition is limited to on-line systems (such as those that may be found on PDAs) or to specific applications (such as mail sorting or check reading) where the vocabulary is quite limited.

Thus, a new segmentation method was developed for finding words and lines in unconstrained handwritten documents. This method is based on an iterative and ascending process that groups connected components into text lines, as explained in the next paragraph (complementary details can be found in [11,14]). Because of the nature of the corpus (which was mostly composed of handwritten modern texts) we did not use character recognition techniques (as these techniques are inappropriate for handling such documents). It should be noticed however that OCR techniques could be useful for processing well-written hand-printed medieval texts.

The line segmentation method works as follows. First, connected components which have a reliable direction are chosen as directional anchor points. Each anchor point constitutes the starting point of a possible alignment. Alignments are formed by linking neighboring components to anchor points by perceptive criteria which are proximity, similarity and direction continuity. The quality of each alignment is then evaluated. Alignment conflicts are solved by applying a set of rules which consider the configuration of the alignments and their quality. After a few iterations the complete alignments appear. At each iteration, the proximity constraint is relaxed in order to link more distant components. The computed lines are then enclosed with the approximated graphic tracks as shown in Figure 6.

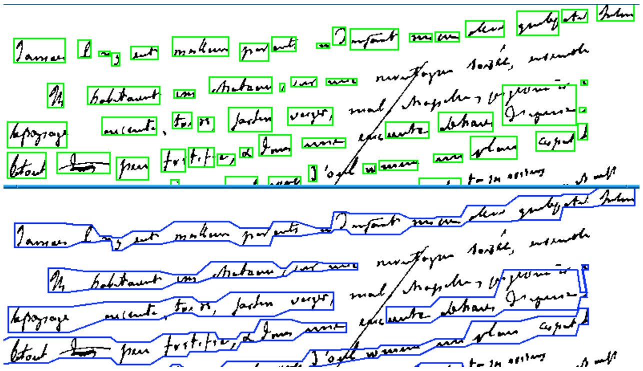


Figure 6: Automatic line segmentation in manuscript images.

3 XML / TEI ENCODING

Besides interactive text/image coupling, another leading design principle of our system was the conception of an open architecture. This made the choice of the encoding scheme quite critical as it should offer a high level of standardization while making it possible to provide powerful text/image coupling mechanisms. For this reason, we decided to base our encoding system on the XML/TEI, a well-known markup language that has been widely used for the humanities. This scheme has several advantages:

1) A preexisting encoding format such as the TEI defines a "common ground" that makes various researchers able to share data and work together. Using such a clearly defined format facilitates the sharing of resources produced with our editing tool as it makes literary materials reusable by others. Moreover, various users can process and interpret these materials in their own way and at various levels of granularity.

2) This open architecture makes it easy for our system to cooperate with other XML tools. According to this philosophy, a collection of specialized tools can be used together instead of having to deal with one single huge monolithic program (which would quickly become obsolete or not flexible enough for performing certain tasks). Users are thus free to use appropriate specific tools that can evolve separately. As a consequence, we were able to focus our own work on its most original aspects, some unimplemented functions being made available by using other tools.

3) The use of XML also facilitates the distribution of such materials. The tools we developed in this study were first designed for editing hypermedia documents. As will be shown in sections 4 and 5, these tools also offer powerful capabilities for document browsing and hypermedia navigation. However, one could object that the average reader may not need rich but specific tools, but would rather get a better access to document data bases through a Web based solution. XML encoding makes this argument irrelevant as the documents produced by our system can also be used by standard Web-based software. This can either be done by browsing them directly (by using an XML compatible Web browser), or by transforming them into more easily displayable representations (such as HTML pages with related images and maps.) This design (which will be detailed in section 4) lets the user choose the most appropriate tool, depending on the precise task he has to perform (e.g. editing a document or just reading it.)

3.1 Encoding of the text /image correspondance

As indicated previously, the internal representation of manuscripts is based on the XML version [29] of the TEI encoding scheme [23,24]. A given manuscript is represented in our system by means of three complementary files:

- the image file,
- the transcription file,
- the mapping file.

The image file is a GIF or JPEG file. The transcription file is a simple TEI-encoded file where line, paragraph and column are recorded using the **l**, **p** and **cb** TEI elements. Each **l** and **p** elements are associated with an **id** attribute.

```
<text>
<body>
<cb>
<l id="l1">Les yeux s'affaiblirent, les persiennes n'ouvraient plus. Bien des années</l>
<l id="l2">se passèrent; - \& la maison ne se louait pas \& ne se vendait pas.</l>
<p id="p1">
<l id="l3">Dans la crainte qu'on ne la renvoya, Félicité ne demandait</l>
<l id="l4">aucune réparation. Les Lattes du toit pourrissaient. Pendant tout un</l>
<l id="l5">hiver son traversin fut mouillé. Apres Paques, elle cracha du sang</l>
<l id="l6">Alors Lemaire Simon eut recours a un docteur. Félicité voulut savoir</l>
<l id="l7">ce qu'elle avait, mais trop sourde pour entendre, un seul mot lui parvint</l>
<l id="l8">"pneumonie"</l>
</p>
</cb>
</body>
</text>
```

The structure of the mapping file is a little more complicated. It is used to express that a given **l** element is the transcription of a given written portion in the image. However, because image fragments are contained in a separate file and of course do not bear an **id** attribute, it is not possible to point at them using the simple SGML/XML ID/IDREF mechanism. Therefore, we have to create a kind of textual surrogate for each image portion which is to participate in a transcription link. We do this by creating a TEI **xptr** element for each related image portion. A TEI **xptr** element defines a pointer to another location in the current document or an external document. Attributes associated with **xptr** include **doc** (to specify the document within which the location is to be found) as well as **from** and **to** (these latter attributes specify the start and endpoint of the portion in the document which is pointed to).

```
<ptrGrp>
<xptr id="xp1" type="line" doc="coeur.gif"
  from="space (3d) (0 125 861) (15 205 175)"/>
<xptr id="xp2" type="line" doc="coeur.gif"
  from="space (3d) (0 121 862) (15 253 208)"/>
<xptr id="xp3" type="rect" doc="coeur.gif"
  from="space (3d) (0 121 907) (0 270 491)"/>
<xptr id="xp4" type="line" doc="coeur.gif"
  from="space (3d) (0 852 124) (30 303 372)"/>
</ptrGrp>
```

As illustrated in this example, we use the **doc** attribute to reference the manuscript image that contains the links. The **from** attribute allows to specify the link location inside the image. A link location corresponds to a graphic track that has been created with an interactive marker. The **type** attribute value identifies the type of the used marker. This example contains the encoded representation of four links that point to the "coeur.gif" image (three lines and one rectangle). To represent the coordinates of our graphic tracks we use the space location term which is associated with the **from** attribute. Its purpose is to point at a location by using coordinates in n-dimensional space. It applies to entities which represent graphical or spatio-temporal data which are typically not encoded in SGML, but in one of many specialized graphical formats. It is application-dependant and we used this possibility to store a representation of our graphic tracks. The syntax of the **from** attribute is described in Figure 7.

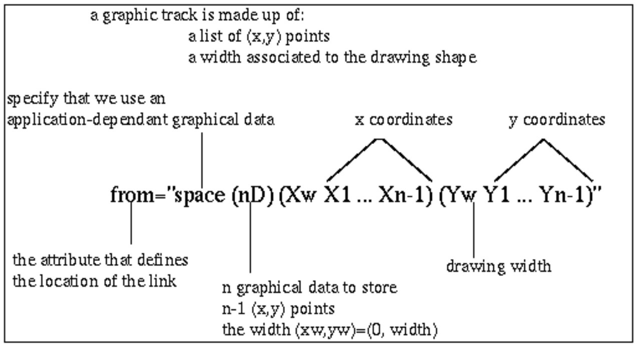

a graphic track is made up of:
a list of (x,y) points
a width associated to the drawing shape
specify that we use an application-dependant graphical data
x coordinates
y coordinates
from="space (nD) (Xw X1 ... Xn-1) (Yw Y1 ... Yn-1)"
the attribute that defines the location of the link
n graphical data to store
n-1 (x,y) points
the width (xw,yw)=(0, width)
drawing width

Figure 7: The encoding of text/image links.

Note that we currently can not rely on the Xlink/XPointer family of specifications because, although very similar in function, for the time being these specifications propose no specific mechanism for pointing to an arbitrary sequence of pixels in an image (or more generally, for addressing a specific subpart in a non-structured document).

Once **xptr** elements have been created for each relevant fragment in the image, the correspondence between **l** and matching **xptr** elements is stored by means of the TEI **corresp** element. Related **corresp** elements are then grouped within a **correspGrp** element. A sample mapping file is shown below:

```
<correspGrp>
<corresp targets="l1 xp1"/>
<corresp targets="l2 xp2"/>
<corresp targets="p1 xp3"/>
<corresp targets="l5 xp4"/>
</correspGrp>
```

4 RENDERING

As introduced in the previous section, document rendering can be performed in two different ways. It is either possible to use the advanced rendition facilities of the transcription/editing tool or to send the TEI representation to an XSL-based conversion module as shown in Figure 8. In the latter case, text and image aspects are *loosely coupled* by using the XML/TEI-based representation, so that a set of formatting objects (e.g. HTML elements) has to be derived from this loosely coupled representation. In the former case text and image aspects are *tightly coupled* because they are integrated in a single internal tree structure which allows complex interactions between text and image representations.

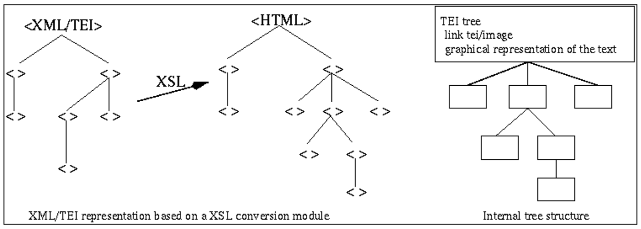

<XML/TEI>
<HTML>
XSL
TEI tree
link tei/image
graphical representation of the text
XML/TEI representation based on a XSL conversion module
Internal tree structure

Figure 8: Two XML/TEI representation methods.

4.1 XSL-based display

XSL (Extensible Stylesheet Language) [30] is a member of the XML family of technical recommendations. While XML aims at describing the logical structure of documents, XSL provides a transformation language (XSLT)[1]. The two halves can function independently of each other, which makes it possible to convert instances of a given document model into instances of a different document model. One of the most common uses of the XSL transformation language is to convert an XML document into a well-formed HTML file (Figure 8).

In an XML transformation, an XSLT processor reads both an XML document to be processed and an XSL stylesheet. Rules found in the stylesheet tell the XSLT processor in which order to traverse the source XML document (which is not necessarily the parse order) and how to react when a particular portion of the source is encountered. For example, when converting an XML/TEI document, the XSL stylesheet can contain rules that instruct the XSLT processor to output a **b** HTML tag each time an **emph** element is encountered during the traversal of the XML document.

We now turn our attention to the kind of XSLT rules we wrote in order to render our XML representation of a manuscript in a Web browser[2]. We chose to convert our representation into a client side HTML map where each area is linked to a script displaying the relevant transcription line. When loading the XML file representing

[1] XSL also includes a formatting language that we do not use here.

[2] Recall from the previous sections that, apart from the image file, it consists of two XML files : an XML transcription file and an XML file containing linking elements that relate lines of the transcription to portions of the bitmapped image of the manuscript.

the manuscript and its transcription, the user is shown the bitmapped image of the manuscript. Then she can click on a line in the bitmap and the matching text is displayed in a separate window (Figure 9).

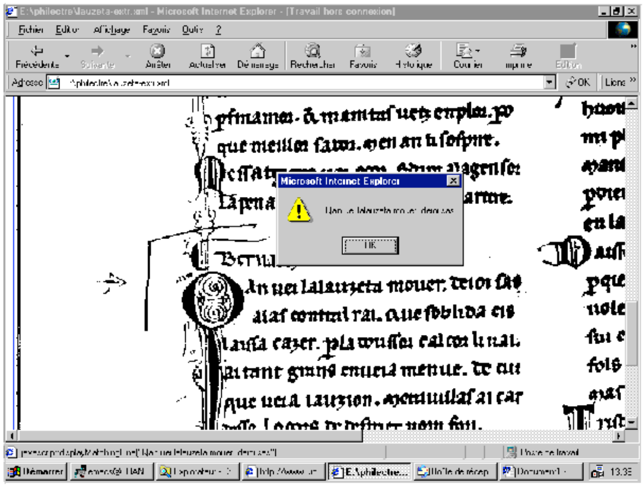

E:\philectre\lauzeta-extr.xml - Microsoft Internet Explorer - [Travail hors connexion]
Fichier Edition Affichage Favoris Outils ?
Précédente Suivante Arrêter Actualiser Démarrage Rechercher Favoris Historique Courrier Imprimer Edition
Adresse
OK
Liens
Microsoft Internet Explorer
OK
Démarrer

Figure 9: Rendering in a Web browser.

To achieve this, our XSL template rules generate an area element for each **xptr** element encountered in the XML representation. The computed value of the **href** attribute associated with each area is the invocation of a *Javascript* function whose argument (the text of the matching line in the transcription) is dynamically retrieved during the translation process. For example, given the XML input below:

```
..............................................
<l id="l2">Qan uei lalauzeta mouer. deioi sas</l id="l2">
<l id="l3">alas contral rai. que soblida eis</l>
..............................................
<ptrGrp>
  <xptr id="img_l2" doc="lauzeta.gif" type="rect"
      from="space (3d) (0 250 650) (0 1385 1405)"/>
  <xptr id="img_l3" doc="lauzeta.gif" type="rect"
      from="space (3d) (0 250 650) (0 1425 1445)"/>
</ptrGrp>
```

We produce the following HTML map where each area has been associated with a dynamically built Javascript function call:

```
<img src="lauzeta.gif" usemap="#transcription">
<map name="transcription">
<area shape="rect"
    href='javascript:displayMatchingLine("Qan uei lalauzeta mouer.
    deioi sas")'
    coords=250,1385,650,1405"/>
<area shape="rect"
    href='javascript:displayMatchingLine("alas contral rai. que
    soblida eis")'
    coords=250,1425,650,1445/>
</map>
```

It should be remembered from the previous sections that our representation includes a set of **xptr** and **corresp** elements. Each **xptr** acts as a surrogate for a portion of the image which has been transcribed. **corresp** elements have two attributes, source and target, that relate a text line of the transcription (identified by the source attribute) with a **xptr** (identified by the target attribute). Our XSLT rules use this information: for each **xptr** they create an area element and they use the information found in the related **corresp** element to dynamically generate a Javascript function call whose argument is the transcribed text related to the **xptr**.

4.2 Unifying XML and graphical representations

The transcription/editing tool (that was described in section 2) also provides advanced capabilities for browsing hypermedia documents (Figures 3 and 11). This tool implements a tight text/image coupling scheme: it relies on an original architecture that makes it possible to integrate XML encoding and GUI components in a single representation.

This architectural design is made possible by using a new graphical toolkit that was developed at our institute as a separate project. This GUI toolkit, called *Ubit* [12,13] is based on the concept of *basic bricks*, which are lightweight modular objects that can easily be combined together. This design, which mimics toy construction games, offers multiple advantages. First, it gives great flexibility so that user interface objects can really match the needs of the application domain. This is because the generic structure of these basic bricks makes them easy to derive into a large variety of customized objects that can fit various specific tasks. So, it is for instance possible to define Ubit bricks that implement the functions of HTML or XML elements.

Secondly, the use of very lightweight components makes it possible to achieve elegant and fully object-oriented architectures without sacrificing performance. In contrast, most graphical toolkits provide UI components that implement many standard features (even when useless for a specific task) and, consequently, tend to use large amounts of memory. This approach imposes serious limitations on the number of UI objects that can reasonably be used. As a consequence, efficient programs tend to mix hybrid programming styles and to introduce various "programming tricks" that considerably increase software complexity and limit its capabilities.

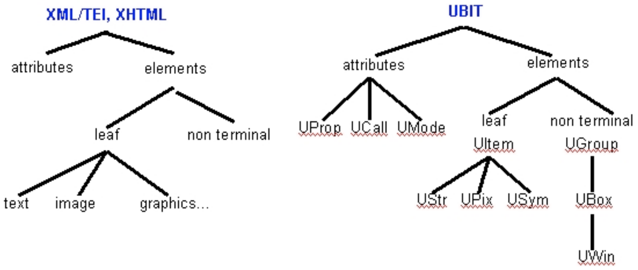
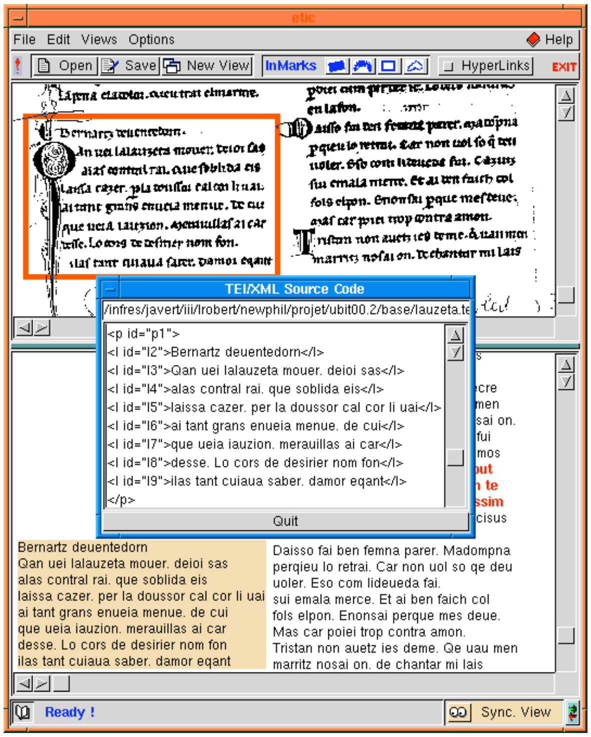


Figure 10: XML / Ubit node correspondence

We took advantage of the specific features of the Ubit toolkit in the following way. Instead of dealing with separate internal representations for the XML tree and the set of UI objects, we merged them into a single graph (Figure 10). More precisely, UI objects are directly mapped to the XML tree in that sense that each XML node is a Ubit brick that implements both appropriate XML/TEI and graphical functions. This homeomorphic representation is well suited to building interactive editors, as each tagged object appearing on the screen effectively corresponds to an actual (and unique) object in the internal representation. So, the textual representations shown on the screen (Figure 11) can be seen as a *projection* of the internal XML/Ubit tree. This tree is automatically managed (and updated on the screen) by the Ubit toolkit and can be simultaneously displayed as several separate (but synchronized) views.



```
<p id="p1">
<l id="l2">Bernartz deuentedorn</l>
<l id="l3">Qan uei lalauzeta mouer. deioi sas</l>
<l id="l4">alas contral rai. que soblida eis</l>
<l id="l5">laissa cazer. per la doussor cal cor li uai</l>
<l id="l6">ai tant grans enueia menue. de cui</l>
<l id="l7">que ueia iauzion. merauillas ai car</l>
<l id="l8">desse. Lo cors de desirier nom fon</l>
<l id="l9">ilas tant cuiaua saber. damor eqant</l>
</p>
```

Bernartz deuentedorn
Qan uei lalauzeta mouer. deioi sas
alas contral rai. que soblida eis
laissa cazer. per la doussor cal cor li uai
ai tant grans enueia menue. de cui
que ueia iauzion. merauillas ai car
desse. Lo cors de desirier nom fon
ilas tant cuiaua saber. damor eqant

Daisso fai ben femna parer. Madompna
perqieu lo retrai. Car non uol so qe deu
uoler. Eso com lideueda fai.
sui emala merce. Et ai ben faich col
fols elpon. Enonsai perque mes deue.
Mas car poiei trop contra amon.
Tristan non auetz ies deme. Qe uau men
marritz nosai on. de chantar mi lais

Figure 11: The transcription/editing tool. The XML/TEI encoding that corresponds to the manuscript image and its transcription is shown in a separate window.

Among other advantages, this design provides *reverse correspondence* from the visual representation to the internal data. This means that the correspondence between cursor location on the text and XML nodes is permanently known. Thus, moving the cursor on the screen is conceptually equivalent to moving a pointer in the XML/Ubit tree. This feature offers extended editing capabilities and various other interesting functions, as the user can directly interact with a visual representation that is virtually equivalent to the internal data.

This approach also makes it possible to create various synchronized views in a simple way. XML/Ubit bricks do not only integrate XML semantics and appropriate text rendition functions. They also include the spatial data that makes it possible to link the content of manuscript images with their textual counterpart. XML/Ubit nodes thus integrate three different kinds of data in a single internal representation:

- the XML characteristics of the tag, its attributes, and its textual content,
- the graphical features for appropriate rendition of the tagged text in the transcription and XML view,
- the spatial coordinates of this text in the manuscript image.

This single internal representation can then be simultaneously *projected* on several GUI views. These views can filter and parameterize the projection mechanism in order to obtain appropriate rendition. One view could for instance display a reduced overview of the whole manuscript image, another view could let the user zoom on manuscript subparts, while other views would display the transcription or the XML encoding (Figure 11).

However, this tight correspondence between XML tagged nodes and their graphical representations (both in the manuscript image and in the text views) might seem surprising at first glance, as XML encoding tends to separate the structure and semantics of the text from its graphical rendition. First, it is important to remember that this tight correspondence exists in the internal representation of the data only in order to improve data management and user interaction, but does not have any impact on external (XML/TEI) representations. Second, another interesting point is that standard graphical characteristics for text rendition are not stored in XML/Ubit nodes but in separate *Style objects*. Style objects, another standard feature of the Ubit toolkit, make it possible to parameterize UI brick classes in a simple and efficient way. Once again, this feature facilitates the translation between the XML and the UI worlds, thanks to a double level of correspondence:

- the XML tag / Ubit Brick correspondence for text representation,
- the XML style sheet / Ubit Style Object correspondence for text rendering.

5 HYPERMEDIA NAVIGATION AND BROWSING

Literary work does not come as a single isolated document but consists of a large set of heterogeneous data (manuscripts, ancient editions, commentaries, etc.). For example, as far as ancient works are concerned, the relationships between the many handwritten manuscript variants produced by copyists since the Middle Ages should be represented. Similarly, as regards modern works, the specialists may wish to relate the many layers of writing that are to be found on the same page. An efficient hypermedia system is thus required to browse and visualize this complex data.

Visualizing and exploring large information spaces are difficult tasks. Users get quite often "lost in the hyperspace". They can not figure out precisely where they are in the information space and which path they should follow to get rapid access to the data they are looking for (as so many navigation paths are usually possible.) Besides, users often have a hard time remembering how to come back to previous interesting locations.

This disorientation problem is quite common when dealing with large information spaces [17]. Among other reasons, it is caused by the fact that users can only see a small fraction of the entire hypermedia at a given time (typically, the page that is currently displayed by the browser) and have no knowledge of its global structure. Thus, in a hypermedia system the representation of the document is critical to allow users to navigate more efficiently. Consequently, we have developed tools for visualizing both the organization of the hypermedia and the context of a detailed document.

The disorientation problem is dealt with in several ways in our system. First, it is possible to open as many transcription windows as needed. As the screen would then quickly become cluttered, these representations are scalable in order to let the user organize his workspace in a way that reflects his current focus of interest. Typically, the focus window (the transcription the user is currently working on) will be displayed at full scale while a small set of interesting or related pages will remain shown at a smaller size. This design favors idea associations as several subparts of the hypermedia can be displayed simultaneously.

The context of the focus window is shown through a global representation of the hypermedia structure. It is quite obvious however that an exhaustive representation would be inefficient as literary hypermedia are generally too large and too complex to be entirely displayed. We thus developed several *focus+context* techniques [8,20]. These representations make it possible to represent the current focal point (e.g. the page the reader is working on) and its local context simultaneously. The following subsections briefly present some two- and three-dimensional techniques that were developed for this purpose.

5.1 Perspective Views

The Perspective Wall paradigm [15] has been adapted for visualizing literary corpus which contains many *versions* or *variants* of the same manuscript pages: several variants of manuscript pages are generally available in the case of ancient works (such as medieval *chansonniers*), while authors often produced several successive *versions* of most pages in the case of modern works. The visualization model follows this implicit two-dimensional structure as shown in Figure 12, where pages are represented horizontally on the wall, while variants or versions are represented vertically (so that a given column contains all the possible variants or versions of a given page). Pages are represented on the wall by means of icons which are linked to the corresponding transcription windows. These visualization tools are synchronized so that selecting a given icon on the wall automatically opens the corresponding page in the transcription browser and vise-versa.

Two different rendering techniques have been used for displaying the perspective view. The left side of Figure 12 illustrates the three-dimensional rendition scheme (which was implemented by using the OpenGL toolkit) while its right side shows a two-dimensional technique that simulates the perspective effect.

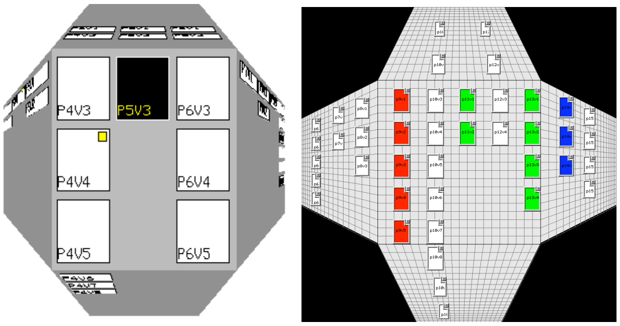


Figure 12: Perspective views

5.2 Links and Visual Bookmarks

The perspective wall model was also extended so as to represent hyperlinks. Links that are anchored in a given page are displayed on the "back side" of the wall when this page is selected (Figure 13). This feature, which makes it possible to show relationships between pages, is only available in the 3D model as it requires the user to move his point of view in order to see the back side of the wall.

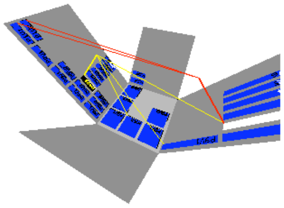


Figure 13: The back side of the wall

Finally, we also developed complementary techniques for browsing and organizing data in a contextual way [19]. For instance, the information workspace shown in Figure 14 was designed for helping users to retrieve documents they previously used. In this system, which consists in a hierarchy of visualization planes, documents are represented by visual bookmarks that can be freely placed and rescaled by the user. This capability makes it possible to implicitly relate the level of details of a document with the user's level of interest (according to the rule that the more an object is large, the more it is visually attractive).

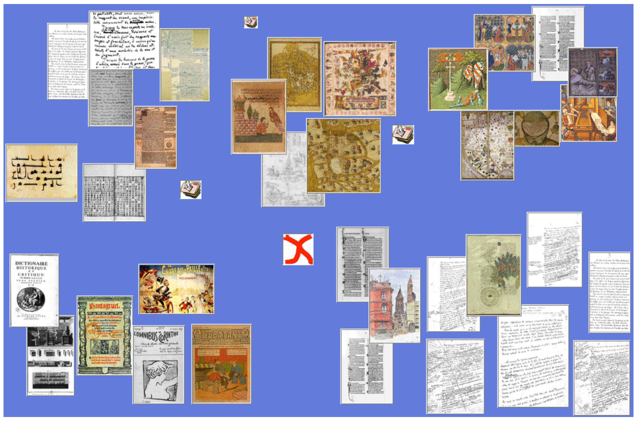


Figure 14: An information workspace based on visual bookmarks

6 CONCLUSION

We have presented a system devoted to the handling of complex literary works with rich handwritten and printed sources. This system unites several layers of information (image, logical structure) in order to provide the user with global perception of the work. Several specialized tools have been designed for this purpose. This contrasts with the systems mentioned in the introduction, most of which are limited to linking flat texts and facsimile images by using proprietary or very specific formats[3].

Another key feature that distinguishes our system from other environments is that it tries to integrate and unify techniques generally developed in separate research communities (image analysis, structured documents and visualization techniques). Furthermore, as pointed out in previous sections, this attempt to couple several domains was performed by designing an open architecture with various modules that rely on a non-proprietary encoding scheme. This design scheme makes it possible for various tools to collaborate and evolve separately. It also facilitates the sharing and the distribution of literary materials. Moreover, this flexible design lets users work with appropriate tools that meet their real needs (for instance, usual XSL-based tools versus specific editing and browsing programs.)

Finally, it is interesting to notice that this work is also related to the concept of *active reading*. This new research domain which tends to attenuate the traditional separation between the editing and reading phases is especially promising, because new specialized *reading appliances* such as e-books and e-ink are now becoming available at reasonable prices [22]. This aspect opens up new perspectives that could be of great interest for image-based humanities computing.

ACKNOWLEDGMENTS

We would like to thank Jean-Louis Lebrave, Pierre-Marc de Biasi, Bernard Cerquiglini and the other researchers from ITEM and IRHT (CNRS) that took part in the Philectre project. We are also grateful to Jacques André, Jean-Daniel Fékete and Hélène Richy for valuable discussions. We also thank Laurence Likforman-Sulem who developed the automatic segmentation software. Lastly, this project was initially made possible by a grant of the GIS "Sciences de la Cognition".

REFERENCES

1. J. André and J-D. Fekete and H. Richy. Mixed text/image processing of old documents. *Cahiers GuTenberg*, no. 21, p. 75-85, 1995.
2. N. Blake, P. Robinson, E. Solopova. The Canterbury Tales Project. http://www.shef.ac.uk/uni/projects/ctp/index.html.
3. Editors and Staff of the William Blake Archive. The Persistence of Vision: Images and Imaging at the William Blake Archive. *RLG DigiNews 4.1.* http://jefferson.village.virginia.edu/blake/main.html.
4. R. Bouché. Debora. Rapport technique, Université de Lyon I, 2000.
5. A. Bozzi, S. Calabretto. The Digital Library and Computational Philology. *Proc. of the European Conference on Digital Libraries*, p.269-285, Oct. 1997.
6. A. Bozzi. Pour un système de philologie numérique. *Document Numérique : les documents anciens*, vol. 3, no. 1-2, 1999.
7. The Electronic Beowulf Project. The Electronic Beowulf Home page. http://www.uky.edu/~kiernan/eBeowulf/guide.htm.
8. S. K. Card, J. D. Mackinlay, B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman 1999.
9. S. C. Hu, R. Furuta, E. Urbina. An electronic edition of Don Quixote for humanities scholars. *Document Numérique : les documents anciens*, vol. 3, no. 1-2, June 1999.
10. Inote : an image annotation tool in Java. http://jefferson.village.virginia.edu/inote/.
11. E. Lecolinet, F. Role, L. Likforman-Sulem, L. Robert, J-L. Lebrave. An Integrated Reading and Editing Environment for Scholarly Research on Literary Works and their Handwritten Sources. *Proc. of the ACM Conference on Digital Libraries.* p. 144-151, June 1998.
12. E. Lecolinet. The Ubit Toolkit Home Page. http://www.enst.fr/~elc/ubit.
13. E. Lecolinet. A Brick Construction Game Model for Creating Graphical User Interfaces: The Ubit Toolkit.

[3] For example, in the BAMBI system the data (raw transcription and mapping information) is exported in the form of HyTime documents, while our system relies on the more widely recognized TEI encoding scheme.

Proc of the IFIP Conference on Human-Computer Interaction (INTERACT), p. 510-518, Sept. 1999.

14. L. Likforman-Sulem, C. Faure. Extracting lines on handwritten documents by perceptual grouping. *C. Faure, P. Keuss, G. Lorette, A. Winter (Eds) : Advances in Handwiting and drawing: a multidisciplinary approach"*, p. 21-38, Europia Press, 1994.
15. J.D. Mackinlay, G.G. Robertson, S.K. Card. Perspective Wall: detail and context smoothly integrated. *Proc. of the ACM CHI Conference*, p. 173-179, 1991.
16. C.C. Marshall, Toward an Ecology oh Hypertext Annotation, *Proc. of the ACM Conference on Hypertext*, p. 40-57, June 1998
17. J. Nielsen, G.G. Robertson, S.K. Card. *Hypertext and Hypermedia*. Academic Press, 1990.
18. Perseus Digital Library Home Page http://www.perseus.tufts.edu/
19. L. Robert, E. Lecolinet, Digital Annotation and Exploration Techniques for handling Image-Based Hypermedia. *Proc of the IFIP Conference on Human-Computer Interaction (INTERACT)*, July 2001.
20. R. Rao et al. Rich Interaction in the Digital Library. *Communications of the ACM*, vol 38-4, p. 29-39, 1995.
21. B. N. Schilit, M. N. Price, G. Golovchinsky, Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. *Proc. of the ACM CHI Conference,* p. 249-256, June 1998.
22. B. N. Schilit, M. N. Price, G. Golovchinsky, K. Tanaka, C. C. Marshall, As We May Read: The Reading Appliance Revolution. *IEEE Computer*, vol. 32, no. 1, p. 65-73, Jan. 1999.
23. C. M. Sperberg-McQueen, L. Burnard. Guidelines for Electronic Text Encoding and Interchange (TEI P3). *ACH-ACL-ALLC Text Encoding Initiative*, 1994.
24. The Text Encoding Initiative. Text Encoding Initiative home page. http://www.tei-c.org.
25. Gusnard de Ventabert (collective name). Représentation et exploitation électronique des documents anciens numérisés. *Document Numérique: les documents anciens*, vol. 3, no. 1-2, juin 1999.
26. J. Virbel. Reading and managing texts on the Bibliotheque de France station. *Text based computing in the Humanities*, p. 31-52, MIT Press, 1993.
27. The Vergil Project. http://vergil.classics.upenn.edu/project/
28. World Wide Web Consortium, Document Object Model (DOM) Level 1 Specification, *W3C Recommendation*. http://www.w3.org/TR/REC-DOM-Level-1.
29. World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition), *W3C Recommendation*. http://www.w3.org/TR/2000/REC-xml-20001006.
30. World Wide Web Consortium, XSL Transformations (XSLT) Version 1.0, *W3C Recommendation*. http://www.w3.org/TR/xslt