A brief introduction to the GeM annotation schema for complex document layout

John Bateman
University of Bremen
Bremen, Germany
bateman@uni-bremen.de

Renate Henschel
University of Stirling
Stirling, Scotland
rhenschel@uni-bremen.de

Judy Delin
University of Stirling *and*
Enterprise IDU
Newport Pagnell, England
judy.delin@enterpriseidu.com

Abstract

In this paper we sketch the design, motivation and use of the GeM annotation scheme: an XML-based annotation framework for preparing corpora involving documents with complex layout of text, graphics, diagrams, layout and other navigational elements. We set out the basic organizational layers, contrast the technical approach with some other schemes for complex markup in the XML tradition, and indicate some of the applications we are pursuing.

1 Introduction

In the GeM project ("Genre and Multimodality": http://www.purl.org/net/gem)[1] we are investigating the relationship between different document genres and their potential realizational forms in combinations of text, layout, graphics, pictures and diagrams. The central focus of the project is to develop a theory of visual and textual page layout in electronic and paper documents that includes adequate attention to local and expert knowledge in information design. By analysing resources across visual and verbal modes, we aim to reveal the purpose of each in contributing to the message and structure of the communicative artefact as a whole. We see it as crucial, however, that research of this nature be placed on as solid an empirical basis as has become common in other areas of linguistic inquiry. The data basis for many of the claims made in this area hitherto has been far too narrow: the provision of suitable corpus materials is therefore fundamental.

For such an enterprise to succeed, it is essential to obtain or construct a structured set of data on which to base the analysis; that is, in the words of Gunther Kress, a leading researcher in the area of multimodal meaning (cf., e.g., Kress and Van Leeuwen (2001)): we need "to turn stuff into data". In our case, the "stuff" consists of raw paged-based information presentations, such as illustrated books, newspapers (print and online versions), instructional texts, manuals, and so on; the "data" is then highly structured re-representations of these documents that bring out parallel but interrelated dimensions of organization crucial for the total effect, or meaning, of the 'page'.

Although it is (a) widely accepted that data for designing and improving natural language processing can best be made available in the form of structured, standardized annotated corpora and (b) increasingly accepted that such data should stretch to include more than the traditional concerns of linguistics–i.e., speech and plain text data–and take in more visually challenging presentations, movements in this direction have to date been very limited. The GeM annotation scheme is being developed in order to support analyses of the broader range of layout-text-graphical interactions that is commonplace in professionally designed documents. We are currently annotating an exploratory corpus in order to bring out the complex interrelationships that can be observed within page-based information delivery.

2 Annotation content

The starting basis for our annotation draws on some detailed non-computational accounts of the organization of multimodal pages/documents–most specifically, the seminal account of constraints on document design by Waller (1987)–and exploratory computational accounts–such as the layout structures

[1]The GeM project is funded by the British Economic and Social Research Council, whose support we gratefully acknowledge. We also thank the anonymous reviewers for this workshop for some very useful comments.

introduced by Bateman et al. (2001). This organization reflects both artefact-internal considerations such as the layout, text and graphics, as well as artefact-external considerations such as design decisions, production constraints (e.g., cost), and artefact constraints (i.e., the limited size of a piece of paper contrasted with the theoretically unbounded scrollable window on a computer screen). These external considerations are often connected. The 'ideal' layout of information on a page might as a consequence never occur: it must be 'folded in' to the structures afforded by the artefact, and labelled and arranged according to the structures required for access.

In order to pick apart and explicitly represent the strands of meaning that we believe play a crucial role in multimodal page-based document design, we require several orthogonal layers of annotation. We claim that these levels are the *minimum* necessary for revealing accounts of the operation of the kinds of visual artifacts being gathered in our corpus–we expect further layers to be added. Indeed, we consider it a crucial design feature that the annotation layers adopted be additive and open rather than excluding and closed. The layers at the focus of attention within the current phase of the GeM enterprise are:

- Rhetorical structure: the rhetorical relationships between content elements; how the content is 'argued';
- Layout structure: the nature, appearance and position of communicative elements on the page;
- Navigation structure: the ways in which the intended mode(s) of consumption of the document is/are supported.

We then need in addition to these layers, explicit representation of constraints that range freely over the layers and which relate design decisions to document types, or genres. Further constraints that are known to determine document design include: *canvas constraints*, arising out of the physical nature of the object being produced (e.g., page or screen, fold-geometry in leaflets, and so on), *production constraints*, arising out of the production technology, and *consumption constraints*, arising out of the time, place, and manner of acquiring and consuming the document. Further details and background for our approach to document design and description are given in Delin et al. (2002).

Our corpus needs to contain information about each of these contributing sources of constraint in a way that supports empirical investigation. Our hypothesis, following Waller (1987), is that not only is it possible to find systematic correspondences between these layers, but also that those correspondences themselves will depend on specifiable aspects of their context of use. But to verify (or otherwise) this hypothesis, the data gathering and annotation must come first. And this leads directly to some important technical issues, since the structures induced by these layers of constraint can be highly divergent and need to be mapped onto one another with extreme flexibility. The well-known corpus annotation problem of intersecting hierarchies therefore arises here with considerable force.

3 Technical approach

Our approach to implementing the required multiple layer annotation scheme is to adopt multiple level 'stand-off' or 'remote' annotations similar to those suggested by Thompson and McKelvie (1997) or the Corpus Encoding Standard (e.g., CES, 1999: Annex 10). For each document to be included in the corpus, therefore, we create a 'base level' document whose purpose is provide a common set of units to which all subsequent stand-off levels refer. These base level units range over textual, graphical and layout elements and give a comprehensive account of the material on the page, i.e. they comprise everything which can be seen on the page/pages of the document, including: orthographic sentences, sentence fragments initiating a list, headings, titles, headlines, photos, drawings, diagrams, figures (without caption), captions of photos, drawings, diagrams, tables, text in photos, drawings, diagrams, icons, tables cells, list headers, list items, list labels (itemizers), items in a menu, page numbers, footnotes (without footnote label), footnote labels, running heads, emphasized text, horizontal or vertical lines which function as delimiters between columns or rows, lines, arrows, and polylines which connect other base units. Each such el-

ement is marked as a base unit and receives a unique base unit identifier.

The more abstract annotation levels may then group base units as required; these groupings must again be very flexible–for example, it is quite possible that non-consecutive basic units need to be grouped (and that differing non-consecutive basic units will be grouped within differing annotation layers). Each of the more abstract layers is represented formally as a further structured XML specification whose precise informational content and form is in turn defined by an appropriate Document Type Definition (DTD).[2] Each layer defines a particular structural view of the original document. The markup for a single document then consists minimally of the following four inter-related layers:

Name	content
GeM base	base units
RST base	rhetorical structure
Layout base	layout properties and structure
Navigation base	navigation elements and structure

All information apart from that of the base level is expressed in terms of pointers to the relevant units of the base level. This stand-off approach to annotation readily supports the necessary range of intersecting, overlapping hierarchical structures commonly found in even the simplest documents.

The relationships of the differing annotation levels to the base level units is depicted graphically in Figure 1. This shows that base units (the central column) provide the basic vocabulary for all other kinds of units and can, further, be cross-classified.

Space precludes a detailed account of the organization of all the levels of the annotation scheme. Instead we select some examples of an annotated document at each layer of annotation to give an indication of the annotation scheme in action. For further technical details and specifications of the annotation scheme, the interested reader is referred to the technical manual (Henschel, 2002). For ease of exposition, we will draw most of our examples from the annotation of the page shown in Figure 2. This page has

[2]For the DTDs themselves, as well as further information and examples, see the GeM corpus webpages.

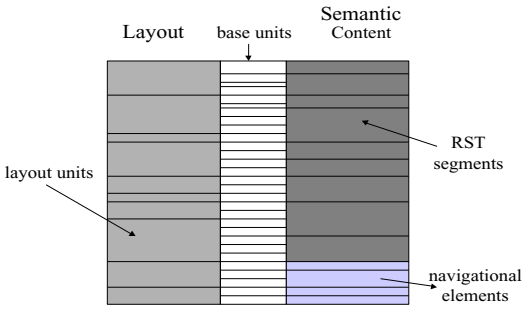

Figure 1: The distribution of base elements to layout, rhetorical and navigational elements

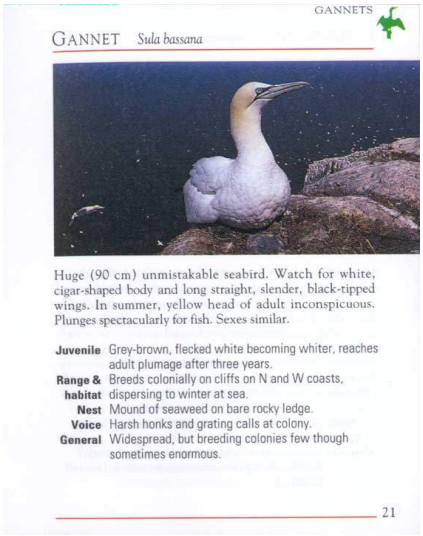

Figure 2: Example page: Flegg, J. (1999) *Collins Gem Birds.* Italy: Harper Collins. p21. Used by kind permission of the publisher.

the advantage that it is relatively straightforward; more complex examples can be found on the corpus webpages.

The base unit annotation (GemBase) for an extract from the centre of the page is as follows:

```
<unit id="u-21.5">---------------</unit>
<unit id="u-21.6"
  src="gannet.jpg" alt="gannet-photo"/>
<unit id="u-21.7">
 Huge (90cm) unmistakable seabird.
</unit>
<unit id="u-21.8">
  Watch for white, cigar-shaped body and
  long straight, slender, black-tipped wings.
</unit>
<unit id="u-21.9">
  In summer, yellow head of
  adult inconspicuous. </unit>
```

```
<unit id="u-21.10">
  Plunges spectacularly for fish.</unit>
<unit id="u-21.11">Sexes similar.</unit>
```

Although the base annotation generally has a flat structure, in certain cases, we diverge from this and allow nested markup, i.e., base units inside base units. This is used in the following situations: emphasized text portions in a sentence/heading, icons or similar pictorial signs in a sentence, text pieces in a diagram or picture, arrows and other graphical signs in a diagram or picture, and document deictic expressions occurring in a sentence.

The layout base then consists of three main parts: (a) layout segmentation–identification of the minimal layout units, (b) realization information – typographical and other layout properties of the basic layout units, and (c) the layout structure information–the grouping of the layout units into more complex layout entities. Whereas in typography, the minimal layout element (in text) is the glyph, here we are concerned with groupings of base units that have a visual coherence and unity with respect to the organisation of the page: these groupings are termed layout units and, unlike the base units, are organized into a non-trivial hierarchical structure as required to describe the page. Again, each layout-unit has an **id** attribute, which carries an identifying symbol; in addition, however, the stand-off annotation is achieved via an attribute **xref** which points to the base units which belong to that layout unit. It is possible, but not necessary, to store the corresponding text portions of the original text file between the start and end tag of a layout-unit for mnemonic purposes; this text information is not used in any further processing. The following extract shows the layout unit corresponding to the main block of text underneath the gannet photo.

```
<layout-unit id="lay-flegg-text"
   xref="u-21.7 u-21.8 u-21.9
         u-21.10 u-21.11">
   Huge (90cm) unmistakable seabird.
   Watch for white, cigar-shaped body
   and long straight, slender,
   black-tipped wings. In summer, yellow
   head of adult inconspicuous. Plunges
   spectacularly for fish. Sexes similar.
</layout-unit>
```

The second part of the layout base is the realization. Each layout unit specified in the layout segmentation has a visual realization. The most apparent difference is which mode has been used – the verbal or the visual mode. Following this distinction, the layout base differentiates between two kinds of elements: textual elements and graphical elements marked with the tags <**text**> and <**graphics**> respectively. These two elements have a differing sets of attributes describing their layout properties. The attributes are generally consistent with the layout attributes defined for XSL formatting object and CSS layout models. The **id** of each layout unit of the segmentation part of the layout base has to occur exactly once under **xref** in the realization part: either in a <text> or a <graphics> element. In the following coding example, we have five layout units which share typographical characteristics. These correspond to the five table cells in the first column of the table on the page shown in Figure 2.

```
<text xref="lay-21.12 lay-21.14 lay-21.16
            lay-21.18 lay-21.20"
      font-family="sans-serif"
      font-size="10" font-style="normal"
      font-weight="bold"
      case="mixed" justification="right"
      color="black"/>
```

The third part of the layout base then serves to represent the hierarchical layout structure. Generally we assume that the layout structure of a document is tree-like with the entire document being the root; this will certainly be problematic for some document types but also has sufficient applicability to enable us to make considerable headway. Each layout chunk is a node in the tree, and the basic layout units, which have been identified in the segmentation part of the layout base, are the terminal nodes of that tree. In our annotation, we use several different tags for the nodes in the layout tree. The three most common are: <**layout-root**>, the element describing the entire document, <**layout-chunk**>, all non-terminal nodes in the layout tree except for the root, and <**layout-leaf**>, the terminal nodes. A slightly simplified (i.e., some further substructure is ommitted) extract of the layout structure for our example page is depicted graphically in Figure 3; it is described by the following XML annotation:

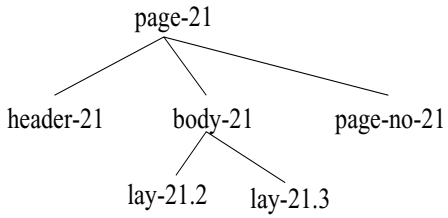

Figure 3: Example page layout structure shown graphically

```
<layout-root id="page-21">
      <layout-leaf xref="header-21"/>
      <layout-chunk id="body-21">
         <layout-leaf xref="lay-21.2"/>
         <layout-leaf xref="lay-21.3"/>
      </layout-chunk>
      <layout-leaf xref="page-no-21"/>
</layout-root>
```

Whereas the annotations so far specify the hierarchical structuring of a page into visually distinguishable and groupable layout units, we need also to record specific information about how layout units are placed on their pages: the page or page segment layout is not fully determined by grouping layout units into a tree structure since further information is required about the actual position of each unit in the document (on or within its page). For this, we introduce an **area model**, which serves to determine the position of each layout-chunk/layout-leaf in an abstract, but fully explicit, way. The area model is a generalization of common notions of 'page models'. Each page usually partitions its space into sub-areas and these can be used for positioning or aligning layout units or subtrees of the layout structure. For instance, a page is often designed in three rows – the area for the running head (row-1), the area for the page body (row-2), and the area for the page number (row-3) – which are arranged vertically. The page body space can itself consist of two columns arranged horizontally. These rows/columns need not to be of equal size. For the present, we restrict ourselves to rectangular areas and sub-areas, and allow recursive area subdivision. The partitioning of the space of the entire document is defined in the **area-root**, which structures the document (page) into rectangular sub-areas in a table-like fashion.

The tag to represent the area root is <**area-root**>. The tag to represent the division of a sub-area into smaller rectangles is <**sub-area**>, this shares the attributes of the root but adds a **location** attribute so that subareas are positioned relative to their parent. Locations are indicated with respect to a logical grid defining rows and columns. The area model for our example page therefore contains a single column with 5 rows (the header, the photograph, the text block, the table, and the footer), in which the fourth row, the table, is itself made up of a subarea corresponding to the rows and columns of a virtual table. This is captured by the following annotation:

```
<area-root id="page-frame" cols="1" rows="5"
      hspacing="100" vspacing="5 40 15 45 5"
      height="16cm" width="14cm">
  <sub-area id="table-frame" location="row-4"
      cols="2" rows="5" hspacing="10 90"
      vspacing="100"/>
</area-root>
```

The attribute `vspacing=''5 40 15 45 5''` means that the area for the running head takes 5% of the entire page height, the area for the next row 40%, etc. The area model then provides logical names for the precise positioning of the layout units identified in the hierarchical layout structure. This makes it straightforward to indicate, for example, that collections of siblings in the layout structure share (or fail to share) some particular alignment properties within the page.

The RST base presents the rhetorical structure of the document. The rhetorical structure is annotated following the Rhetorical Structure Theory (RST) of Mann and Thompson (1988). The relation between rhetorical structure and layout is currently an important area of study in multimodal document description and so its inclusion in the GeM annotation scheme is essential. Several annotation schemes for RST have been proposed in the literature (cf. Daniel Marcu and Mick O'Donnell's proposals, e.g., www.sil.org/~mannb/rst/toolnote.htm and the RAGS rhetorical level: Cahill et al. (2001)). The precise GeM notation differs from these in certain respects, but the main principles of representation remain similar. Since many of the details of the rhetorical annotation presume some familiarity with the decomposition of texts according to RST, we will note here simply that

this annotation layer again represents a hierarchical decomposition in which the leaves of the tree can correspond to both textual and graphical base units.

Finally, the navigation base captures those parts of the document/page which tell the reader where the current text, or 'document thread', is continued or which point to an alternative continuation or continuations. These make up the navigation layer of annotation. The addresses used by such pointers are either names of RST spans or names of layout chunks. For long-distance navigation, typical nodes in the RST structure and in the layout structure have been established for use in pointers; in particular, chapter/section headings are names for RST spans and page numbers are names for page-sized layout-chunks. This structure imposed by the navigational elements is thus quite different from the preceding layers and can freely cross-cut the hierarchicies expressed there. Again, for further details, the reader is refered to the documentation manual.

4 Comparison with other XML-based approaches

It is useful to consider other approaches to representing 'overlapping hierarchies' that have been proposed in the XML literature; since these are early days in the construction of such annotation schemes, it is likely that the experience gained with differing schemes will prove highly beneficial for further development.

The first examples of extensive overlapping hierarchies within markup for NLP are probably to be found in speech corpora. It is clear that intonational phenomena, for example, may or may not respect grammatical or other kinds of structure and so need to be maintained separately. Speech-oriented corpora generally use the time line as a basic reference method since speech events are necessarily strictly ordered in time. This is quite different in the GeM case where we have found the non-linearity and the non-consecutive nature of the units grouped within our annotation scheme as presenting a major problem for annotation models that have been developed in the speech processing tradition where contiguity of units is the expected case. Whereas the speech signal can be encoded by means of time-stamps, in the GeM model we need to use the layout structure (or even the area model within the layout structure) instead for placing elements within the physical document.

One of the most detailed general considerations of the range of XML-based solutions to the multiple, overlapping hierarchies problem, as well as an extensive listing of further literature, is given by Durusau and O'Donnell (submitted). After reviewing several requirements and approaches to the problem, Durusau and O'Donnell propose a *Bottom-Up Virtual Hierarchy* approach which (i) creates multiple instances of the source document, each with its own consistent XML-defined hierarchy and (ii) creates one further document instance called the 'base file', in which each basic element of the document is linked via XPath expressions to its position in each of the separately defined hierarchy documents. The position in each separate hierarchy is captured as the value of a distinct attribute to a base element tag. This means that the base file becomes extremely complex, although Durusau and O'Donnell envisage that this file could in future be automatically constructed and maintained.

The similarity of this approach to the independently developed GeM approach argues to a certain extent for the necessity of proceeding in this direction. The differences between the approaches arise from the different tasks that are being considered and the corresponding differences in emphasis. Durusau and O'Donnell are considering the task from the perspective of differing 'interpretations' of a text in the sense more commonly pursued in text corpus markup: it is, therefore, natural to consider each hierarchy as an autonomous marked-up document. We are concerned with linguistic analyses of documents, where the *structure* of the analyses themselves is itself a major focus of attention. It is then no longer so important that each level of analysis transparently represents a 'view of the text'. When querying the corpus for structural and realizational regularities, we are free to do this across any set of the annotation layers present, using the full power of properly parsed XPath expressions, and without the need to always decompose such queries into the terms of the elements of the base file. This is the XML reflex of the linguistic strategy

of stratification of linguistic information across the linguistic system.

Single structured text views can, of course, be created out of the GeM markup by following the indirection links present in any individual GeM annotation layer. This requires that that layer be interpreted with respect to the corresponding base level document and is not then 'locally' complete. This (formally slight) complexity is, we feel, more than balanced by the fact that we need neither any additional complexity in our base level markup nor the double coding of the position of nodes in hierarchies and base elements. It is also straightforward to introduce intermediate levels of structural analysis–as, for example, already done in our selection of base units as units relevant to layout rather than 'words' or other more straightforward formal units. Indeed, when the more traditional linguistic levels of annotation (syntax, semantics) are added into the complete scheme, many of the GeM annotation layers will continue not to access the leaves of these structures at all, and will continue to work with the base units illustrated here; this variable granularity may prove to be a general requirement for linguistically complex analyses. And, finally, we also need not recompile our base level document whenever an additional layer of annotation is added to the scheme, thereby simplifying maintenance. Further comparison of the approaches will, however, require more detailed evaluation in use.

Finally, we can consider the GeM approach as contrasted with directions within the XML community itself since there, too, there are proposals for capturing distinctions of content, layout (e.g., in terms of formatting objects: XSL-FO) and navigational elements (e.g., in terms of Xlink). Whereas we are attempting to make the GeM description as compatible with these constructs as possible–for example, as noted above with respect to the realizational possibilities for the layout units–it is important to understand the very different aims involved. The purpose of the GeM project is to analyse the multimodal decisions made in a wide range of document types and it is not yet clear which theoretical levels and which theoretical constructs within those levels will prove appropriate. The formatting object description is only suited to a certain class of layout types (which excludes, for example, newspapers) and so is in many respects too specific for our more exploratory purposes. We are also searching for effective levels of abstraction at which to characterize our data: effective here meaning that these will be the constructs over which canvas constraints, production constraints and consumption constraints are most appropriately expressed. Perhaps, to offer an NLP analogy: whereas the XML modelling decisions correspond to a fine-scaled phonetic description of a language event, we are in the GeM project searching for the higher levels of abstraction corresponding to the grammar, semantics and pragmatics of the language events. We expect this to give us a substantially better theoretical grasp of the meaning-making potential of layout decisions and their control by external constraints.

5 Applications of the annotated corpus

A number of uses are currently being made of the annotated GeM corpus. While our empirical study will need considerably more data to be encoded before we can make reliable statements concerning the patterning of various constraints with document decisions, we have already been struck by the rather wide variation that exists within single documents between selected layout structures on the one hand and rhetorical organization on the other. In surprisingly many cases, this variation goes beyond what might be considered 'good' design: in fact, we would argue that most such designs are flawed and would be improved by a more explicit attention to the rhetorical force communicated by particular layout decisions. This represents the use of the corpus for document critique and improvement (cf. Delin and Bateman (2002)).

We are also using the data gathered so far to inform the design of a prototype automatic document generation system capable of producing the kinds of variation and layout forms seen in our corpus. In this work, the annotation scheme provides skeletal data structures that define target formats of various stages of the generation process. Thus, for example, layout planning needs to produce a structure that is an instantiated version of a layout structure as we have defined it above. Some first results are reported

in Henschel et al. (to appear) which describes a prototype implemented as a set of XSLT transformations that convert a content representation into an XSL-FO document. The transformations are conditionalized so as to respond to various features of the content, the modes of the available material, and the rhetorical structure; so far pages generated include further examples of the kind used as an example in this paper as well as pages from instructional texts such as telephone user guides.

Conditionalization is expressed in terms of XPath specifications that check the presence or absence of particular configurations within any of the GeM annotation layers as required. Such specifications are, however, somewhat cumbersome for more complex queries. Whether further developments such as XQL or XQuery will bring benefits is not yet clear. Somewhat disappointing was the unsuitability of the previous generation of linguistic-oriented corpus tools, which, despite considerable investment, seem to have been outstripped by the very rapid developments seen in the mainstream XML community. Most of our current work is done directly with XMLSpy and XLST tools such as Xalan.

The final goal of our corpus collection work and the prototype document generation systems is to place the commonly quoted aim of using XML markup for document 'repurposing' on a solid theoretical foundation. Thus, for example, the ability automatically to generate very different presentational forms for the instructional texts or bird pages mentioned above is an inherent feature of the GeM model. More important for us is to uncover as precisely as possible the conditions which make certain presentational selections more appropriate than others. In general, we relate the need for presenting information in different forms to the kinds of constraints we introduced above: very different canvas constraints are imposed, for example, depending on whether the delivery medium is across the telephone, on a palmtop, or as a display on a big screen. However, it is not simply a matter of the differing affordances of the display device. The selection of particular information and information display modes is also a matter of established document types. These document types change over time, due both to changing production constraints and to the establishment of new genres. It is not possible to deploy inappropriate realizations for established genres; a newspaper that changed its presentation style to that of birdbooks would quickly be out of business–and *vice versa*. Mapping out these possibilities and showing what larger patterns hold is then our eventual goal. And for this, extensive and detailed empirical studies of the kind we hope the GeM corpus and annotation scheme will support are crucial.

References

John A. Bateman, Thomas Kamps, Jörg Kleinz, and Klaus Reichenberger. 2001. Constructive text, diagram and layout generation for information presentation: the DArt$_{bio}$ system. *Computational Linguistics*, 27(3):409–449.

Lynne Cahill, Roger Evans, Chris Mellis, Daniel Paiva, Mike Reape, and Donia Scott. 2001. Introduction to the RAGS architecture . Available at `http://www.itri.brighton.ac.uk/projects/rags`.

Corpus Encoding Standard. 2000. Corpus Encoding Standard. Version 1.5. Available at: http://www.cs.vassar.edu/CES.

Judy L. Delin and John A. Bateman. 2002. Describing and critiquing multimodal documents. *Document Design*, 3(2). Amsterdam: John Benjamins.

Judy Delin, John Bateman, and Patrick Allen. 2002. A model of genre in document layout. *Information Design Journal*.

Patrick Durusau and Matthew Brook O'Donnell. submitted. Implementing concurrent markup in XML. *Markup Languages: Theory and Practice*.

Renate Henschel, John Bateman, and Judy Delin. to appear. Automatic genre-driven layout generation. In *Proceedings of the 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*, University of the Saarland, Saarbrücken.

Renate Henschel. 2002. GeM annotation manual. Gem project report, University of Bremen and University of Stirling, Bremen and Stirling. Available at http://purl.org/net/gem.

Gunther Kress and Theo Van Leeuwen. 2001. *Multimodal discourse: the modes and media of contemporary communication*. Arnold, London.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Henry S. Thompson and D. McKelvie. 1997. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe '97*.

Robert Waller. 1987. *The typographical contribution to language: towards a model of typographic genres and their underlying structures*. Ph.D. thesis, Department of Typography and Graphic Communication, University of Reading, Reading, U.K.