AsTeR –Towards Modality-Independent Electronic Documents

T. V. Raman
Digital Equipment Corporation
Cambridge Research Lab
One Kendall Square, Building 650
Cambridge, MA 02139
Voice-mail: 1 (617) 621-6637
E-mail: ⟨raman@crl.dec.com⟩
WWW: http://www.research.digital.com/CRL/personal/raman/home.html

March 10, 1995

Abstract

The advent of electronic documents and the consequent creation of digital libraries —vast repositories of electronic information— has a profound impact on how we produce, organize, store, retrieve and consume information. All of these activities have been dictated to the present by the technologies used to share information; A change in the underlying technology, namely, the move from paper to electronic documents, offers a unique opportunity to revolutionize how information is archived and disseminated. This paper will focus on a specific aspect of the opportunities opened up by electronic publishing on the NII —the ability to present information in multiple modalities and thereby free it from any single presentation medium.

Traditional printed communication relies on a passive intermediary, paper,for the exchange of information between the author and reader. Ideas put down on paper come back to life only when perused by the reader.

Electronic publishing is mediated by a computer, an agent capable of processing the information. As a consequence, the ideas expressed by an author need no longer be bound to any single "display" form; nor does it require human intervention to translate the information from one displayed form to another. Electronic information can be processed and displayed in a manner best suited to each individual's needs. Thus, the advent of electronic documents makes information available in more than its visual form —electronic information can now be display-independent.

Traditionally, an electronic document has been viewed simply as digitally representing (or the means towards producing) the printed page.

Instead, we view the electronic document as the basic entity that represents information; we allow the information to be rendered in different ways —on paper, spoken, processed in different ways by a computer, etc. This change of viewpoint has allowed us to develop AsTeR (Audio System For Technical Readings) a computing system that *audio formats* electronic documents to produce *audio documents.* AsTeR can speak both literary texts and highly technical documents that contain complex mathematics. Moreover, the listener can ask to have parts of a document repeated in different ways: a document has many different spoken views.

The adequacy of the audio rendering depends on how well the electronic document captures the essential internal structure of the information. In this paper, we discuss capturing structure and give guidelines for authors to follow to ensure that their documents exhibit structure adequately.

In the context of the NII, the digital libraries of the future can be viewed as large information servers that allow multiple clients to access and display information in a format chosen by the user. By obviating the need to move physical media, e.g., printed paper or recorded tapes, the NII enables the ready dissemination of multimodal renderings of information.

1 Introduction

The advent of electronic documents and the consequent creation of digital libraries has a profound impact on how we produce, organize, store, retrieve and consume information. All of these activities have been dictated to the present by the technologies used to share information; A change in the underlying technology, namely, the move from paper to electronic documents, offers a unique opportunity to revolutionize how information is disseminated. The same electronic document can be printed, spoken, spoken in outline form over telephone lines or the Internet, processed automatically to extract certain kinds of information, and so on. This paper will focus on a specific aspect of the opportunities opened up by electronic publishing —the ability to present information in multiple modalities and thereby free it from any single presentation medium.

But for all this to be realized, the electronic document has to be considered as the key component, not the printed page. The electronic document is not the representation of the printed form; the printed form is *one* representation of the electronic document. This means that the electronic document has to be written to convey explicitly as much structure as possible –and details of any one presentation medium, such as the spacing between paragraphs on a printed page and length of time between speaking sentences, have to be abstracted out of the electronic encoding.

Information present in traditional printed documents comes to life only when it is perused by the human reader. Intelligent processing of such information therefore requires explicit human intervention. Intelligent processing — computing— can range from performing symbolic calculations on mathematical expressions occurring in a document, to translating the information to alternative display formats, e.g., audio, hypertext etc. To give a specific example, it requires a trained reader to make printed information available in spoken form[1]

Electronic communication, on the other hand, is mediated by an information processor rather than passive pieces of paper. This means that we can separate out the capture and storage of information from its presentation. Markup systems[2] like (LA)TEX capture the logical structure of a document along with its content. Rendering or presentation —the process of producing a "display"— can be viewed as applying a specific set of transformations to the abstract logical structure encapsulated by the encoding.

Typically, the structure is visually formatted to produce visual layout, a rendering attuned to the eye's ability to rapidly access different parts of a two-dimensional display. Thus, visual rendering projects the document logical structure on paper in a form that enables the reconstruction of the structure envisioned by the author.

Before getting into details of aural presentation, it will be useful to talk about the difference between printed and spoken documents. The passive printed document is processed by an active reader, who can view it in many different ways —read only section titles, skip a piece of mathematics, temporarily skip to a different page to read a referenced theorem, reread an interesting passage, and so on. Such active processing becomes even more flexible when the document appears on a computer screen, because hypertext and calculational capabilities can be used.

When it comes to audio, on the other hand, the *document* is the active player and the human the passive one. The speaker (perhaps on an audio cassette) actively reads in a relentlessly linear fashion, from beginning to end, and the listener simply listens, with little control over the process. Further, producing audio documents can be a laborious and time-consuming task —just ask organizations like Recordings For the Blind (RFB), who are engaged in producing such " talking books"[3].

ASTER –Audio Documents

ASTER (Audio System For Technical Readings) [Ram94] is a computing system that *audio formats* electronic documents to produce *audio documents.* Audio formatting produces renderings that are attuned to an auditory display. In its interactive mode, ASTER changes the active-passive relationship described

[1] Organizations like Recordings For the Blind (RFB) have been engaged in producing such "talking books" —an extremely laborious and time-consuming process.

[2] To most people, *markup* means an increase in the price of an article. Here, "markup" is a term from the publishing and printing business, where it means the instructions for the typesetter, written on a typescript or manuscript copy by an editor. Typesetting systems like (LA)TEX have these commands embedded in the electronic source. A *markup language* is a set of means (constructs) to express how text (i.e., that which is not markup) should be processed, or handled in other ways.

[3] An audio recording of the author's PhD thesis produced by ASTER is the first computer-generated talking book to be produced by RFB.

above by enabling *interactive listening.* The listener can browse the document structure and can obtain different *audio views* of (pieces of) the document.

The interested reader can experience an interactive demonstration[4] of the audio renderings produced by AsTeR on the WWW (available from the author's home page). It aptly brings out the power of the Internet in publishing multimedia documents; none of my journal publications come with online demonstrations. It also emphasizes the display-independent nature of electronic documents; both the audio formatted version and the visually laid out Postscript were generated from the *same* LaTeX source.

We envision digital libraries as repositories that serve information. Unlike libraries of today that store information in a single display format, the digital library of the future could potentially provide customized views of information. The rest of this paper will focus on the generation of multiple views of information objects. We emphasize that such multiple views can be multimodal, i.e., renderings may be visual, aural, and in the general case, a combination of both visual and aural views.

2 Representing Information

All information has structure, and any physical rendering of a document is a projection of this structure onto a particular medium, e.g., printed paper. A "rendering" of a document on some medium is best understood if it makes this logical structure readily apparent. For example, a visual rendering —onto a two-dimensional medium like paper— may use cues like boldface, different fonts, and indenting to help reveal structure. A visual rendering takes advantage of the eye's ability to rapidly access different parts of a two-dimensional display. An audio rendering has to use an entirely different set of cues to reveal structure.

Early in the development of AsTeR, we realized that the ability to render information in a variety of output modalities would be a direct function of the richness of the internal representation used to capture structure and content. Abstractly speaking, the high-level structure of a document is independent of any particular mode of display, and the internal representation should reflect this. As a first step in realizing AsTeR, therefore, we developed high-level models to represent document structure. For instance, the richness of the representation used by AsTeR completely frees the order in which subterms in an equation are rendered aurally from the order in which they would appear on paper. (See 4 for details.)

This section briefly outlines some of the representations used in AsTeR. Rendering this high-level representation is outlined in Section 4. Based on these ideas, we define a set of requirements in Section 5 that should prevent electronic encodings from being tied down to any single display form.

[4]This hypertext document presents a collection of math examples rendered in audio by AsTeR and in Postscript by LaTeX/DVIPS.

Ordering the Possible Representations

The amount of structural information that can be extracted from the electronic source depends entirely on how the logical structure is marked up. In the context of OCR-based document recognition, this is also a function of the quality of the visual rendering being recognized. In the case of both markup-based and OCR-based document recognition, the type of structure that can be extracted varies widely.

Intuitively, there is a hierarchy of document types (lattice) ordered by the amount of structural information captured, and the ease with which such structure can be recognized. The amount of structural information varies from plain paragraphs and sentences marked up with normal punctuation, all the way up to highly technical documents with footnotes, equations and references. The ease with which the structure can be extracted ranges from the bitmap on a low-resolution fax, through to a postscript[5] or PDF[6] file, on upward to a highly marked up LaTeX or SGML file. Given a document instance, the amount of structural information determines which of these logical structures we can extract. Given a plain ASCII document, structural information has to be inferred from the layout of the text, e.g., spacing, vertical alignment and centering. This is also true of pure visual layout encodings like PostScript and PDF. In the case of encodings in markup languages like (LA)TeX, much of the logical structure is explicitly present in the electronic source. Structure based document encoding systems like SGML provide the potential for extracting the richest possible logical structure, since they separate the layout process from the encoding of the document structure.

Document Models in AsTeR

The recognizer used in AsTeR captures logical structure present in documents encoded in the TeX family of languages. An important feature of this recognizer is that it works on the entire gamut of encodings, ranging from plain ASCII documents, i.e., no explicit markup, up to documents containing completely unambiguous encodings of the logical structure.

The basic document model used in AsTeR is the attributed tree. Each hierarchical level of the document is modeled as a node in this tree. Each node can have content, children and attributes. Using object-oriented terminology, each different kind of node of the tree is called an *object*. Thus, "chapter", "section", "paragraph", and "sentence" are all objects. If a document contained five sections, its representation in AsTeR would have five instances of object "section". This object-oriented terminology is used because AsTeR actually uses CLOS objects in this fashion. The use of an object-oriented language was instrumental in allowing us to develop and implement the ideas in AsTeR incrementally and effectively.

[5] PostScript is a registered trademark of Adobe Systems Incorporated.
[6] PDF, Portable Document Format, is a registered trademark of Adobe Systems.

$$\boxed{\begin{array}{ccc} \text{left-superscript} & \text{accent} & \text{superscript} \\ \nwarrow & \uparrow & \nearrow \\ & \textbf{math object} & \\ \swarrow & \downarrow & \searrow \\ \text{left-subscript} & \text{underbar} & \text{subscript} \end{array}}$$

Figure 1: A math object with attributes. Each of the attributes themselves contain math objects.

This attributed tree structure is augmented to represent mathematical content; we call this augmented representation the *quasi-prefix form*, (see figure Figure 1 on page 6). Expressions that are completely unambiguous, e.g., $x + y$, are captured in their prefix form. In addition to linearizing the underlying tree structure, mathematical notation uses *visual attributes* such as superscripts and subscripts, whose interpretation is context-dependent. We extend the prefix form to capture such visual attributes —hence the name *quasi*-prefix.

A key feature of the quasi-prefix form is that it delays the assignment of semantic interpretation to instances of ambiguous written mathematics. At the same time, it is sufficiently rich to permit renderings that are independent of the order in which the written symbols would appear on paper. Linear renderings with the rendering-order hard-coded into the system can be produced with a simpler representation, e.g., a linear list, or even the TeX encoding itself. This was shown by TeXTaLK, a string-substitution based program that directly transformed TeX source to produce spoken renderings [Ram91, Ram92].

As an example, assume that `\kronecker`[7] is defined as an infix binary operator. Given the expression

$$a \otimes b$$

encoded as

```
a\kronecker  B
```

we can represent it in the quasi-prefix form by a tree whose root is object *kronecker*, and write rendering rules for object *kronecker* that produce either "a kronecker product b", or "kronecker product of a and b". The former rendering can be produced by TeXTaLK as well, but a simpler list-like representation restricts the system to this one form of rendering.

In producing printed output, one view is sufficient; once the information has been presented visually, a person reading the material can access it in any desired order. But even with visual rendering, different views may be desired. For example, one may wish a view that gives only the table of contents of a paper. Or, for a document that presents an algorithm, one view could give the whole presentation and a second view could present only the overview of the algorithm. See [Lam93] for a discussion on the hierarchical presentation of

[7] TeX does not provide this operator, and it will have to be defined as a macro. We describe how AsTeR is extended to handle such macros in Section 3.

proofs. The linearity of audio makes it essential that ASTER have the ability to present multiple views. Lack of this feature is one of the major shortcomings of books on tape, where the listener is restricted to the one view presented by the person speaking the text. ASTER allows the listener to explore the material the same as a person perusing printed material, and thereby enables *active* listening.

3 Extending Document Logical Structure

A flexible markup system needs to be extensible. This is because it is impossible to enumerate all possible logical structures that different authors might wish to use. This is perhaps one of the biggest shortcomings with SGML, where extending the logical structure requires modifying the Document Type Definition (DTD), a non-trivial task. On the other hand, one of the most powerful aspects of (LA)TeX is its extensibility. It is possible for an author to define the few additional logical constructs needed by a particular document instance using the TeX macro facility, [Knu84, Knu86].

Macros permit the author to abstract away layout details when writing the document. To give an example, the command `\kronecker` is not present in (LA)TeX. An author can extend (LA)TeX by defining

```
\def\kronecker{\raisebox{1pt}{\:\otimes\:}}
```

and then write

```
$ A \kronecker B$
```

The definition for `\kronecker` has extended the markup language, and consequently, the logical structure that can be expressed. LaTeX [Lam86] is itself a good example of how TeX macros can be used to implement a language for encoding document structure. The presence of user-defined macros in documents presents both a challenge and an opportunity for a system like ASTER.

The Challenge

In general, TeX macros can perform any arbitrary computation permitted by TeX, a Turing-complete language. Hence, it is impossible to directly translate the macro expansion into an audio rendering. The TeX primitives are visual layout operators, and translating a TeX macro directly into an audio rendering rule would imply a one-to-one mapping between the visual and audio rendering. As explained in Section 1, visual renderings are attuned to a two-dimensional display, and audio renderings need to be attuned to an auditory display. Further, expanding a TeX macro loses structural information; when all macros in a document have been expanded, only the visual layout remains.

The Opportunity

A proper choice of macro definitions can encode semantic information about the visual layout being used. For instance, an author wishing to use the notation

$\left(\frac{P}{Q}\right)$ to denote the Legendre symbol could define a new command `\legendre` that produces the desired layout. In the absence of this facility, e.g., when using an SGML Math DTD that does not know about Legendre symbols, the author would be restricted to encoding $\left(\frac{P}{Q}\right)$ in terms of the basic operators provided, in this case, the operator that stacks its arguments. Though both encodings would produce the desired visual layout on paper, the encoding using the `\legendre` macro has the advantage that an application can interpret the use of the notation as meaning a Legendre symbol.

Representing Extended Logical Structure

The first step in solving this problem is to represent instances of user-defined macros in our high-level document model. Producing audio renderings (or renderings in other modalities) of such instances is then equivalent to rendering any other object present in the model.

We model macro definitions as introducing new object types. Thus, defining `\legendre` is equivalent to adding object *legendre* to the set of objects present in the document model. A macro definition in (LA)TEX has two parts; the first part declares the macro and its number of arguments; the second part specifies how instances of this macro call are to be displayed.

Translating this to the object-oriented model, the first part of the macro introduces a new object type; the second part is a visual rendering rule for instances of this object. We extend ASTER to handle user-defined macros according to this framework. Hence, for every user-defined macro, the document model is extended by adjoining a new object type, and calls to that user-defined macro are represented by instances of this new object type.

4 Rendering Information Objects

ASTER renders information by applying *rendering rules* to the internal representation described in Section 2 and Section 3. The system of rendering rules used in ASTER and the language in which they are written (AFL —Audio Formatting Language) are described in detail in [Ram94]. In a sense, AFL is to audio formatting as Postscript is to visual formatting, although AFL is a much smaller language.

Here, we show a small example of such a rendering rule for a user-defined macro. In the following, we use CLOS generic function read-aloud. For the present, let us assume that function read-aloud executes the necessary actions to render its argument.

After extending ASTER to process the (LA)TEX macro `\inference`, which is defined as

```
\newcommand\inference[2]{\frac{#1}{#2}}
```

to render instances of calls to `\inference`, we can define

```
(defmethod read-aloud((inference inference))
  "Sample rendering  for object inference."
  (read-aloud (argument 1 inference))
  (read-aloud "implies")
  (read-aloud (argument 2 inference)))
```

Given $\frac{A}{B}$, this produces "A implies B".

If we wished to produce a rendering that inverts the order in which the arguments to macro `\inference` are rendered, we would define:

```
(defmethod read-aloud((inference inference))
  "Renders inference with arguments reversed."
  (read-aloud "We know that ")
  (read-aloud (argument 2 inference))
  (read-aloud "because")
  (read-aloud (argument 1 inference)))
```

which produces "We know B because A".

Switching between these two rendering rules has the effect of inverting a proof-tree! ASTER makes it easy to write several rendering rules for the same object. ASTER also allows rendering rules to be partitioned into rendering *styles*. In an interactive session with ASTER, switching between rendering styles (a collection of rendering rules for different objects) and invoking individual rendering rules can be done with a few keystrokes, making it easy for a listener to obtain many different views of a document.

ASTER derives its power from representing document content as objects and by allowing multiple user-defined rendering rules for individual object types. These rules can cause any number of audio events (ranging from speaking a simple phrase, to playing a digitized sound). The pitch of the voice, the physical head-size of the virtual speaker, the volume, and many other parameters can be changed by rendering rules, making it easy to create sound cues to help display structure.

To give an example of this, the logo for ASTER is

and is produced by (LA)TEX macro `\asterlogo`. After appropriately extending ASTER to recognize this macro, we can define an audio rendering rule that produces a bark when rendering instances of this macro. Thus, the same piece of markup `\asterlogo` produces the picture of Aster[8] when rendered visually, and an appropriate sound[9] when rendered aurally.

[8] Aster is my guide-dog.

[9] The bark is that of a generic dog, Aster is too well trained to bark, and could not therefore be recorded.

This feature was exploited to advantage when producing the audio formatted version of the author's thesis. The dedication page of the thesis contains a large picture of Aster, and the audio formatted version contains a verbal description of the picture, accompanied by the sound of Aster panting in the background. You can listen to this example on the WWW —visit the AsTeR home page[10] and click on the picture of Aster.

Several ideas come together to make all this possible. First, logical structure is of paramount importance —not its display on any one particular medium. The more a document makes structure explicit, the better the document can be displayed on several different mediums.

Next, the use of (LaTeX) TeX macros to encode structure makes it possible to have a system like AsTeR, in which the internal structure can be extended to fit a document. This allows the encoding of the structure in a flexible, uniform, and consistent representation such as an attributed tree, with the addition of the quasi-prefix form for dealing with mathematics.

Finally, providing different rendering rules and styles and a flexible way to switch among them makes it possible to obtain multiple views of a document in an interactive fashion.

5 Conclusion

We conclude this paper with a few guidelines for encoding document content in a display-independent manner. Electronic encodings that adhere to these guidelines will enable multiple uses of the same electronic source. Though the notion of archiving information in its richest possible form is itself not new, we note that such ideas have been exclusively motivated in the past by the need to display information visually. The richest representation for the specific problem of being able to accurately reproduce the visual appearance of information is not necessarily appropriate for *computing* on the information. Visual presentations, as pointed out earlier, are optimized for human consumption, and therefore necessitate explicit human intervention in performing *intelligent* manipulation of the content.

Our work brings a fresh perspective to this issue by addressing the problem of aurally rendering complex information. It points out that the visual presentation that we are all familiar with, e.g., the printed version of this paper, is just one possible view of the information content, not the information itself. This insight leads naturally to the approach used in AsTeR, namely the development of high-level information representation and the rendering of such representations in different modalities.

To ensure a multiplicity of uses, the digital library should archive information in its richest form. Such encodings should be capable of producing high-quality renderings in the various output modalities, e.g., a well-formatted PostScript or PDF file containing high-resolution fonts, audio renderings that exploit the various features of an auditory display, etc. A digital library may choose to archive

[10] *URL* http://www.research.digital.com/CRL/personal/raman/aster/aster-toplevel.html

one or more of the "display" forms in addition to the high-level encodings as a means of optimizing information delivery. However, archiving information in any of these "display" forms is equivalent to archiving information on printed paper. Hence, such "display" representations should not be viewed as a replacement for the high-level encoding.

Retaining the high-level encoding that generates the various renderings will facilitate:

- Linking multiple views of the information.
- Producing additional views of the information.
- Searching the information.
- Computing on the information in as yet unforeseen ways.

The Chicago Journal of Theoretical Computer Science is an online journal to be published in LaTeX (*URL* http://cs-www.uchicago.edu/publications/cjtcs/) and fulfills these ideals. The markup recommended to authors has been carefully designed to abstract out all layout details by the Managing Editor, Prof. Mike O'Donnell, and we hope to aurally render articles from the journal using AsTeR. See [O'D92, O'D93] for a description of the work leading up to this project.

Unambiguous Encodings

The same visual layout may be used to display disparate concepts. Encoding instances of such ambiguous notation by using well-designed markup abstracts out the layout details from the document encoding, and allows an information agent to identify the different concepts correctly. We illustrate this with a concrete (LA)TeX example.

The visual layout of stacking one mathematical object above another, separated by a horizontal line (horizontal rule), could be used in several contexts.

- **Fraction:** $\frac{1+\sqrt{5}}{2}$
- **Inference rule:** $\frac{X=Y,\, Y=Z}{X=Z}$

Using the encoding `\frac{object-1}{object-2}` in both cases makes it impossible to disambiguate between the different interpretations. When the same layout is used to denote different concepts, these should be marked up distinctly.

For instance, in LaTeX, the author could extend the markup language by defining two new macros:

1. `\def{\fraction}[2]{\frac{#1}{#2}}`.
2. `\def{\inference}[2]{\frac{#1}{#2}}`.

Though stated in terms of (LA)TEX, the above requirement can be generalized to any encoding system. It merely states that objects that are semantically distinct but share a common visual representation should have distinct electronic encodings. This is essential in ensuring that such objects can be presented in other modalities, where they may not necessarily share the same displayed representation. More generally, such distinct encodings are also essential if we are to compute on the content encapsulated by the encoding.

Summary

- Avoid using any display-specific format as the principal form of archiving electronic information, e.g., a scanned bitmap image, a PostScript or PDF file (visual rendering) or a digitized recording (audio).
- Avoid use of explicit visual layout in the electronic encoding. For instance, avoid use of `\vskip` in (LA)TEX documents.
- Use distinct markup to encode semantically distinct objects even if they have the same visual layout.
- Use an encoding system that is extensible by the author; this will ensure that the maximum amount of semantic information is captured at the encoding stage. This minimizes the amount of guesswork that has to be done later.

Electronic document encodings have not always followed these rules, since the markup was viewed purely as a means of producing the visual rendering. Our work points out that the same encoding can be put to multiple uses; it is therefore important to apply principles of good software design and reuse to document encodings as well.

To draw an analogy, we do not currently throw away the program source code once we have successfully compiled it into a running executable; equivalently, it is important to retain the high-level document encodings that produce the final display form in which information is disseminated.

References

[Knu84] Donald E. Knuth. *The TEXbook.* Addison-Wesley, Reading, Massachusetts, 1984.

[Knu86] Donald E. Knuth. *TEX The Program.* Addison-Wesley, Reading, Mass., 1986.

[Lam86] Leslie Lamport. *LATEX: A Document Preparation System.* Addison-Wesley, Reading, Mass., 1986.

[Lam93] Leslie Lamport. How to write a proof. (94), February 1993. To appear in *American Mathematical Monthly.*

[O'D92] Mike O'Donnell. Electronic journals—scholarly invariants in a changing medium. *Conference on Academic and Professional Journals in the Twentieth Century*, April 1992. Presented by author as discussant for session on "The Future of Journals".

[O'D93] Mike O'Donnell. Issues involved in publishing an electronic journal. *Seminars on Academic Computing*, August 1993. Revised version of paper appeared as University of Chicago Department of Computer Science Technical Report 93-11, July 1993.

[Ram91] T. V. Raman. TEXTALK. *TUGboat*, 12:178, March 1991.

[Ram92] T. V. Raman. An audio view of (LA)TEX documents. *Proceedings of the TEX Users Group*, 13:372–379, July 1992.

[Ram94] T. V. Raman. *Audio System for Technical Readings*. PhD thesis, Cornell University, May 1994. *URL* http://www.research.digital.com/CRL /personal/raman/raman.html.