

## Computing and the understanding of text

*Lillian N. Cassel*

*Robert E. Beck*

*Daniel Hardt*

*Villanova University*

*Pensylvania*

*USA*

### ABSTRACT

A paradigm shift is underway in the needs of newcomers to information technology. The computer and its use have invaded the core of how people learn and work, even how people know. This paper describes a course built on the principle that computing is an integral part of the human experience with text in all its forms, from simple strings of characters to intricately entwined hypermedia documents. We extract advanced topics in computing and make them accessible in a nontechnical introductory computing course conducted within the context of the production, management, storage and understanding of text.

*Main conference themes:* knowledge as a resource

*Educational areas:* higher education

*Study topics:* computer literacy

*Secondary keywords:* algorithms, collaborative learning, hypermedia, interdisciplinary, networks

## INTRODUCTION

### **Computing as a core discipline**

We have argued in previous papers [1, 2] that computing as a discipline is as essential to the understanding of fundamental ideas in many fields as mathematics or natural and physical science. Over the fifty year history of electronic computing machines computing power has doubled every one to two years. Consequently we can now be using computing systems which are at least eight times more powerful than what was available at the last WCCE meeting. To harness this power for the nonspecialist we explore synergistic connections between computing and other disciplines. Our goal is to influence the ways of knowing in nontechnical fields.

We report on a nontechnical introductory computing course built on the clustering of ideas of text and computing which fills a part of the mathematics and computing requirements for all students studying for a Bachelor of Arts degree at our university. The course, Tools for Text, combines literature, linguistics and computing in a profoundly different way foreshadowing new approaches to knowledge for the Twenty-First Century.

### **Computing and text**

There are many ways to think about and organize text: it can be a simple linear string, a hierarchy of sections and chapters or a weblike structure of interconnected segments. Here basic concepts in computer science provide important clarity and insight. In Tools for Text we present some basic data structures and algorithms such as linear orders, trees, networks, regular expressions and finite state machines. These provide the conceptual basis to think about and manipulate text in a variety of ways, including searching for strings, searching for information on the Internet, using hypertext, reading interactive fiction and examining basic ideas in modern theories of grammar. The students were asked to compare several types of text (literature, fiction, poetry, letters, notes, lists) and to consider the potential of computing in influencing these. In what follows we first describe the computer science concepts which formed the basis for the course lectures. Then we describe several perspectives on text which were explored and developed in a laboratory setting. Finally we address the issue of collaborative learning in a computing intensive environment.

## TOOLS FOR TEXT: CONCEPTUAL BASIS

A computer is an invaluable tool for text, both in a practical and an intellectual sense. A key background concept in the course is that of an abstract datatype (ADT): a set of objects and the operations which can be performed upon those objects. The string data type and regular expression notation provide an intellectual basis for confronting the great variety of string matching mechanisms available to the computer user. The tree and network data types provide a basis for dealing with new perspectives on text, such as hypertext, the Internet and the evolving form of hyperfiction. And finally an understanding of computational devices such as finite state machines is essential for a rudimentary understanding of modern theories of grammar. More specifically, these perspectives on text were developed in the following ways:

*Texts as strings:* Text can be viewed as simply a linear string of characters, without any further structure. The course examines a variety of operations on strings, such as substring matching, locating of characters and tallying n-grams. This provides a conceptual basis for the analysis of a variety of familiar practical techniques, including searching a directory for files or searching a word processing document for a string of characters. We introduce the regular expression notation as the basis for these operations. In addition, we introduce the finite state machine which is the computational foundation for the regular expression notation. Text viewed as strings of characters leads to discussion of coding, security, spell checking and grammar checking.

*Texts as trees:* It is traditional to view text as a tree structure breaking it down into chapters which are further divided into sections, paragraphs, lines, words and characters. The abstract definition of a tree is presented and regular expressions for some of the familiar building blocks of text are given such as words, lines and paragraphs. The tree structure while simple and conceptually appealing is a restrictive way of organizing text. For example, a given object must have a unique ancestor and links are not permitted to form a cycle. New perspectives on text seek to remove some of these restrictions.

*Texts as networks:* For many years literary theorists have suggested that text be thought of as a weblike structure of associations and links, rather than a simple linear or hierarchical structure. The data type network captures this notion by removing certain restrictions inherent in the tree structure. This data type is instantiated in the notion of hypertext where a portion of text can be linked to some other portion of text. This provides the basis for graphical interfaces to the Internet such as Mosaic and Netscape. In addition the network concept

illuminates the development of the traditional novel to the evolving form of hyperfiction where the progress of the reader can take a variety of paths through the text.

*Theories of grammar:* Modern theories of grammar are generative; that is, the grammar of a language describes a computing device capable of producing all and only the sentences of the language. This computational perspective is the basis for the 'Chomskyan Revolution' in the study of language. A key argument in the development of this perspective is that a valid grammar of English cannot be a finite state machine; rather, the grammar must be some more powerful computational device [3]. This argument constituted a final topic in the course.

## TOOLS FOR TEXT: LABORATORIES

The topics we discuss in the course relate to many areas of study and illustrate the close ties between the study of informatics and other disciplines. To make these ideas concrete and to clarify their practical significance we use laboratory experiences to complement each of the lecture topics. In this section, we describe some of these laboratories, discuss the importance of discovery as part of the learning experience and describe some errors we made in designing some of the laboratory exercises. Copies of the laboratories reside at

<http://renoir.vill.edu/courses/labs/tools4text/labset>

### Some laboratory examples

*The ADT tree:* Lecture introduction of the abstract data type tree defined the elements of a set and identified valid operations on trees. In the laboratory, the students encountered trees in the context of the DOS directory structure [4]. The root of the tree was the root of the network file system. Various subtrees—a directory assigned to CSC courses, another to each individual student—were clearly significant. Further exploration led to discovery of directories containing various classes of software (editors and word processors, mathematics packages, images, databases, even games). Tree operations introduced in lecture included accessing a node, inserting or deleting a node and traversing a tree. In the laboratory these abstract ideas became respectively accessing a subdirectory or file, creating or deleting a subdirectory or file and generating a full directory listing showing all the subtrees. The nature of the tree structure was emphasized by comparing the ease with which one moved upwards in the structure to the difficulty of moving down, where one needed to know the name of the destination beforehand.

The particular property of the ADT tree which requires there be no cycles became meaningful in the context of the directory structure. The fact that a file does not exist simultaneously in more than one subdirectory is at the same time logical and also clearly a restriction. This leads to the practical computing concept of unique path name, which was shown on a single DOS machine, a local network of machines, and across the Internet for both html files and electronic mail addresses.

*The ADT network:* The ADT network removes the restriction of loop free connection found in trees. In the laboratory there are two implementations of the network structure for the students to explore: information accessible in the World Wide Web (WWW) [5] and the literature format built on hypertext. The WWW first gives meaning to the concept of linkages which may contain loops. The students experiment by following the links which appear in various documents. It does not take long to find a link folding back on itself to form a loop.

In exploring the WWW the meaning of 'one document' blurs as pointers jump to logically associated text which may or may not be in the same file or even reside in the same file system. In these first experiences with hypertext the students can easily see the advantages of such links. They replace the traditional concept of a footnote with greatly expanded access to related material. Students quickly experience the difficulties of navigating this space and realize that the advantages of hypertext are balanced by some disruption of the flow of ideas in the base document.

We then move to the notion of fiction with similar freedom of structure. Now the links are not to support material for an aspect of the topic; instead they provide options for the reader in progressing through the story. This modern form of fiction requires that the reader actively participates in the development of the story; the reader must choose the path to follow, circle back and take another branch and actively explore the story space.

### **The role of discovery**

Exploring and discovering is a core component of the laboratory portion of the course. In encouraging experimentation in the laboratory we accomplish two goals: students learn that experimenting is a legitimate way of learning in a computer environment [6] and they learn how to construct a search or other experiment in order to increase the learning opportunities. By emphasizing exploration we open up the opportunities for new discoveries long after the student has completed the course.

The use of discovery in the laboratory is not without difficulties. The challenge is to design and experiment in such a way that discovery will happen when the student carries out the instructions, even though the discovery may not be the one the instructor planned. If the instructions on the laboratory try to 'guide' the student to one particular result, the effect may be to cut off exploration and replace it with yet another precisely specified set of steps. In our first attempt at these laboratories, we fell victim to this trap on occasion. The laboratory shown in the Appendix was a particularly effective invitation to explore.

### **Role of collaboration**

An important aspect of the discovery approach to learning is collaboration during experimentation. Our laboratories are designed to encourage fearless computing, in part by allowing the students to support one another in their explorations. For students who might be fearful of technology, a collaborative approach is beneficial. It is crucial in the discovery process and has long term implications for the students' future as well. By encouraging cooperative work, the laboratories prefigured a realistic work environment. A career involves collaborative environments, ill defined problems and learning directed toward a goal. The disparity in this respect between the educational and the work environment has been recognized by several commentators and is clearly described by Denning [7 ,8]. Collaboration in the laboratory bridges this disparity in work rules and goals.

One goal which was explicitly stated for many of the laboratories was to observe the workings of the user interface by noting the changes in state to the system. Such observation is particularly difficult alone. In collaboration with a colleague our students discovered that there was a wealth of information on a Windows interface if one took the time to absorb it all.

The laboratory work on hyperfiction included several aspects of collaboration. Each student read the same hyperfiction short story, but chose a different path through it. To challenge their assumptions about the work and the author's point of view the group of students was asked to list the characters and provide links between them using the computer science structure of the network. Some were amazed to find that the story may have taken place in two different locations, with two unrelated sets of characters. As a second collaborative effort the students communicated with the hyperfiction author through electronic mail.

### **The laboratory experience in summary**

The laboratories are designed to connect fundamental lifelong knowledge of principles of computing with current computing practice. Our emphasis was on experimentation and discovery, not predetermined results. In each

laboratory, one lecture topic formed the motivating force for the particular type of exploration. Our goal was to give the students some high level computer science ideas and to make those ideas accessible in a nontechnical introductory environment. Students who continue with the study of the computing sciences will revisit these ideas in much greater depth.

## CONCLUSIONS

Our universities have been challenged to meet the needs of students for foundational knowledge which can evolve as the worldwide shifts in market, political alliance and economics occur. This knowledge will come through increasingly sophisticated applications of information technology and new styles of working which involve collaboration, access to vast bodies of information and virtually instantaneous communication with colleagues. Our course is one attempt to meet this challenge.

## REFERENCES

1. Beck, R. E., Cassel, L. N., and Austing, R. H. (1989) *Computer science: a core discipline of liberal arts and sciences*. SIGCSE Bulletin. **21** (1) pp. 56-60.
2. Cassel, L. N., Beck, R. E., and Austing, R. H. (1990) *The interdependence of disciplines: computer science as a full partner*. Proc. World Conference on Computers in Education, Sydney, pp. 253-258.
3. Chomsky, N. (1957) *An elementary linguistic theory*. Chapter 3 in Syntactic Structures, Mouton, The Hague.
4. Beck, R. E., Cassel, L. N., and Hardt, D. *Tools for text, laboratory 3: files and directories*. <http://renoir.vill.edu/courses/labs/tools4text/lab3>
5. World Wide Web Project.  
<http://info.cern.ch/hypertext/WWW/TheProject.html>
6. Carroll, J. M. (1982) *The adventure of getting to know a computer*. Computer, **15** (11) pp. 49-58.

7. Denning, P. (1992) *Educating a new engineer*. Comm. ACM, **35** (12) pp. 83-96.
8. Denning, P. (1993) *Designing new principles to sustain research in our universities*. Comm. ACM, **36** (7) pp. 99-104.

## APPENDIX

### CSC 1010—Tools for Text, Spring, 1995, Laboratory 10

#### Hypertext fiction—I have said nothing

*Objective:* Read a short work of hypertext fiction—I Have Said Nothing, by J. Yellowlees Douglas.

Hypertext fiction—hyperfiction—is a new way of reading and writing stories, made possible by the computer. Hyperfiction stories are written with hypertext. Each screen contains one or more links to other screens within the story. The process of reading hyperfiction is interactive. The reader must decide, after reading a screen, where to go next. Conventional fiction is a linear sequence of pages; turning the page is the only choice after finishing a page. Hyperfiction is not linear. Instead, since each ‘page’ (screen) can be linked to many other screens, the structure is a network rather than a line.

*Reading hyperfiction:* Upon finishing a screen, there are two ways to continue: **Enter:** you to the default continuation screen. This is analogous to simply turning the page in a book.

**Links:** Sometimes there is no default continuation screen or you may wish to pursue a different link. To do this, single-click on the **Links** button. You are presented with a list of all the links from the current screen. To select a link, simply double-click on the line describing that link.

**The End:** If you reach a screen called *The End in I Have Said Nothing*, **Enter** won’t work, but you can continue by using the **Links** button. With Hyperfiction the reader, not the writer decides when it’s the end.

*Starting ‘I Have Said Nothing’:* Start Windows. Double-click on the icon *I Have Said Nothing*.

Begin reading I Have Said Nothing. As you are reading, keep a log of each screen that you visit:



*Paraphrase:* As you are reading, answer the following simple questions:

- Who is Sherry? What happens to her?
- Who is Juliet? What happens to her?

*Study questions:* In the screen, *Can we accept one?*, Douglas poses the following question: “How is it we, who have witnessed eighteen thousand deaths, cannot accept one?”

- What are the eighteen thousand deaths being referred to?
- Is Douglas’ question a reasonable one? Why or why not?
- How do you answer it?
- The title of the story is related to the following quote:

“I have done nothing but wish to speak; if I have spoken, I have not said what I wished to say.”

- Where does this quote appear in *I Have Said Nothing*?
- Who is the quote from?
- The quote deals with the relation of speaking and intention: what you want to say *versus* what you actually succeed in saying or communicating. How does this relate to the difference between hyperfiction and conventional fiction?

*The experience of reading hyperfiction:*

- What are the most notable differences in reading hyperfiction *versus* reading conventional fiction?
- Are there other narrative forms that could be compared to hyperfiction?
- Do you like this way of reading a story? Why or why not?
- How do you think the process of writing hyperfiction differs from that of writing conventional fiction?

**Homework:** write a page on one of the three study questions. Submit it to your lab instructor by email by April 20, 1994.

Send an email message to the author, J. Yellowlees Douglas, answering the following questions:

- What did you like about reading the story? Did you find it satisfying? What in particular was satisfying?
- What didn't you like about reading the story? Did you find it frustrating?
- What in particular was frustrating?

In your answers, make sure to refer to specific aspects of the story in describing what you did and didn't like. Also, make sure you send a copy of your message (using the cc option) to your lab instructor.