Discourse Organisation in Computing Science Discourse: A Preliminary Study

Francisco Alonso Almeida-Laura Cruz García-
Víctor González Ruiz
Universidad de Las Palmas de Gran Canaria

The present work aims at describing the structure of computing science texts in a very limited time-span. These texts show a similar organisation of information; and we have also noticed the presence of certain linguistic elements which we consider generic of these texts. We have also concentrated on the visual elements which accompany the main text. These visual elements are classified according to their form.

1. The aim of study

This paper aims at describing the way how information is conveniently organised in computing science popular papers, such as those contained in computer magazines like *Communications of the ACM*, and *Dr. Dobb's Journal*. Our study will focus on highlighting the schematic structure of those papers revealing the grammar frequently associated with those stages in the construction of computing science discourse. A detailed description of the visual aids used to guide the reader through the text will be also offered.

2. Material of study

The material used for this study consists of fourteen articles published in the latest issues of different computer science magazines: *Communications of the ACM, PC Magazine, Dr. Dobb's Journal, Byte* and *C++ Report*. They have an extension which ranges from about 1500 to 4500 words, in order to show that the organisation of information does not necessarily depend on the extension of the text. The articles normally deal with a certain computer product, its advantages and disadvantages, its evolution and the new improvements, and also and quite frequently the similarities and differences between the given product and other products of the same kind.

We have decided to work on the latest issues (from November 1998 until January 1999) of the corresponding magazines. The reason for this choice lies in the fact that we are not very much interested in the evolution of the informative structure of this type of discourse, but in finding features which are common to a certain number of articles belonging to the same period of time, in order to describe the way information is structured and presented and, thus, set a general pattern. The journals selected are directed toward a specific audience formed by computer science experts and users with a certain level of knowledge of the matter.

They are not research papers, since they give general information about a computer product and do not intend to show any research academic study. The authors of the texts are usually editors, professors, programmers and researchers.

3. What's a popular computing science text?

Popular computing science texts basically aims at presenting new information about a specific product, showing how that specific product works, and very often pointing out the advantages of that product with respect to a former one. Obviously, this sort of information is addressed to a specific sort of audience. This audience constitutes a parameter to take into account when defining the concept of popular computing science text. Also, the formal characteristics of these texts suggest the writer's preoccupation to achieve clear messages which the reader may comprehend effortless. Within the formal characteristics, it should be noted the presence of certain typographical elements which guide the writer in his search to find certain pieces of information.

The audience of these texts is basically constituted by users, and people with a certain degree of knowledge. The consequences of this idea is that certain pieces of information should necessarily be ignored, since they may prove redundant for these types of readers that own a wide knowledge of the topic under discussion.

Obviously, this does not imply that the writer includes a background introduction which shows the antecedents to the product which he is about to describing. What it does really mean is that the writer deems unnecessary to describe former products in detail. Would the reader need that sort of omitted information, he should direct himself to books oriented to the explanation of those particular products. Those books normally accompany the purchased product to describe it, or they might be bought at any bookshop.

Apart from conditioning the amount of information to be included in those texts, the audience also conditions the type of language used there. In other words, the writer makes efforts to describe the products using clear language departing from excessive gilding strings, and complicated subordination which may obscure the information the article deals with. This is specially relevant if we consider that many of these texts describe computing programmes which, to some extend, also require the use of mathematical elements which may posit some complications by themselves. So, why drawing the readers' attention to untangle difficult language when the aim of the paper is to illustrate the pro's and, rarely the con's, of new products?

A more formal definition of these texts involves the use of certain grammatical features which appear recursively in the different texts under study. These features which we label as generic of this sort of texts are (1) the use of the present simple tense, (2) expressions of reason, purpose and result, (3) the presence of causative verbs, (4) the use of comparative and superlative adjectives and adverbs, (5) the massive presence of *can*, and (6) the construction *by* + *-ing*. It could be argued the convenience of including in the above list the presence of technical vocabulary in those texts as a way to recognise them. However, we think the specific type of vocabulary used here is also part of other computing science papers, and thus it does not prove generic of these texts.

What follows is a description of the visual devices used in order to guide the reader through the text. Secondly, we offer the schematic structure of the texts studied here, and finally, the recursive linguistic elements in these texts are presented.

3.1. *Visual devices in computing science popular papers*

Visual devices are normally used in this type of texts as a way to highlight specific bits of information, some times. Other times, these devices constitute mere summaries describing some aspects related to the product presented. The nature of these visual devices are varied, but we can summarise those in (1) typographical elements such as bold type-face script, the use of bigger fonts, dropped capital letters, and the like; (2) in-text information boxes such as schemas, drawings, flowcharts, and so on.

3.1.1. *Typographical elements*

Typographical elements are used in our texts to signal how information is divided into different stages so that the reader can find in a reduced amount of space the information they require. For instance, the author of the text *Get the Message* systematically employs the bold type face in order to signal the beginning of the description of a new product:

Telelogos Media Transfer
Media transfer, from the French firm Telelogos, ... (p. 8)
XcelleNet RemoteWare
With over 650,000 users in 1700 organizations, Xcellenet is the leading vendor... (p. 9)
Mobile Objects
Mobile Objects differ from Media Transfer and ... (p. 9)
The Wireless Future
Mobile computer users will increasingly turn to wireless telecommunications... (p. 11)

In the case of other authors, they combine the use of bold-type face with italics as a way to indicate subdivisions of the information, as can be seen in the following example from *The Virtual Design Team*:

The VDT Micro Theory of Project Organizations
Organizational Engineering is the process of configuring an organization structure... (p. 85)

VDT Implementation
The VDT model uses...
Activities. VDT shifts the object of analysis from an aggregated organization...
Communications. Coordination requires information flow...
Actors and information processing. Because of its aggregated view of organizational information...
Allocate attention. Activity subactivities and communications accumulate...
Process information. After an actor selects an activity or coordination item...
Send communications to other actors. Actors use communication...
Generate and handle exceptions. VDT actors generate, communicate, and process several...
Exceptions and decision making. Actor information processing and exception...
Programming implementation. The VDT system was implemented as an object-oriented...
Discussion
As discussed in the procurement policy case study ealier, we used VDT to .... (pp. 85-91)

In the above example, the author uses the bold-type face and italics as a way to signal the division of his text. In doing so, his achievement is twofold: (1) the division of the text into different units of information, and (2) the presentation of his text as reader-friendly. The way he uses those two typographical elements is by

highlighting sections in bold, and subsection in a combination of bold type-face and italics. Further subsections included in subsections are indicated by the use of italics as shown above. This technique also allows the writer to avoid excessive wording by tackling directly those aspects which must be necessarily discussed. This is specially relevant if we consider the small amount of space devoted to each article in each magazine. In other words, articles are only few pages length, what implies that the writer must try to condense all the information he wants to pass on to his readers in clear stages.

3.2.2. *In-text information boxes*

We understand by the concept of *in-text information boxes* to those schemas, figures, drawings, and flowcharts which are included in our articles as a way to clarify certain important points dealt with within the main body text of the articles. Sometimes the in-text information boxes are used to include the stages of *conclusion* and *further information.*
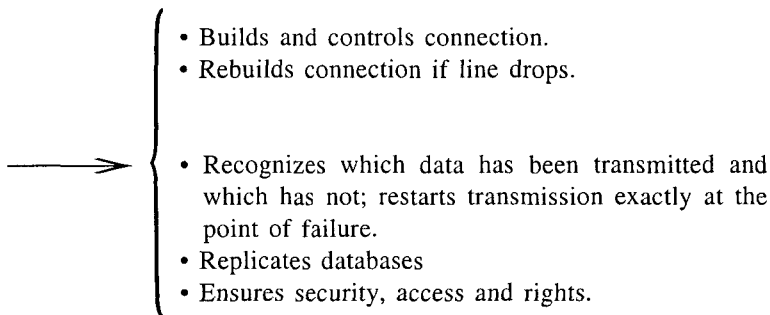
In-text information boxes constitute an undeniable help to the reader, mainly because they can resort to them when they find the message obscure or when they need a drawing to follow the author's explanation. An example of schema is the following (*Get the Message*, p. 8):

Messaging Middleware

Application

Presentation
- Builds and controls connection.
- Rebuilds connection if line drops.

Messaging
Middleware

⟶
- Recognizes which data has been transmitted and which has not; restarts transmission exactly at the point of failure.

Session
- Replicates databases

Transport
- Ensures security, access and rights.

Network

Data Link

In the example above, the writer outlines the qualities of the product in a sort of schema. By doing so, he facilitates the reader's comprehension of the message, partly because the language he uses at this point is far more simple than in the body text. In addition, the presentation of the general features of this product in such a clear outline allows the reader to check in seconds if the product would be of any benefit to him, so that readers can eventually save their time.

Another example of in-text information boxes is found in the same article *Get the message* (p. 8). Here, the author presents a checklist comparing the features of the products he describes in his article. Tables like this help the reader to have a complete view of the products described in the article in a very limited period of time.:

Middleware Features Checklist

Product	Media Transfer	Remote-Ware	Mobile Objects	Nokia WDE
Store and forward	√	√	√	√
Error recovery	√	√	√	√
Data compression	√	√	√	
Syncronization	√	√		
Custom coll. hand.	√			
Subscr. management		√	√	
Asset management		√		
Mixed transports	√	√	√	

√ = yes

4. The structure of texts

4.1. *Division of texts in sections*

Most computing science popular papers contain an almost identical organizational pattern. The reader can find the same textual stages in nearly all the articles under study. However, there exist several varying factors, such as the section of the magazine where the article is in (which determines where the focus must be on), or the author of the text (whose individual style may modify the relatively fixed structure).

A generic pattern of organization for this type of texts is outlined below:

GENERAL TOPIC	4.1.1. NAME OF SECTION = type of information
PRODUCT	4.1.2. TITLE 4.1.2.1. PRE-/POST-TITLE HEADLINES
AUTHORSHIP	4.1.3. AUTHOR (CV)
INFORMATION	4.1.4.1. Introduction 4.1.4.1.1. Purpose of paper 4.1.4.1.2. Contextualization 4.1.4.2. Product description 4.1.4.3. Conclusion/Evaluation 4.1.4.4. Reference to further information

As a matter of illustration, we will apply this pattern to the contents of one of the computing papers described (*Get the Message*):

Name of section: INTERNATIONAL

Title: Get the Message

Post-title headline: A new class of middleware irons out the uncertainties of remote networking.

Author: By Dick Pountain

Introduction: What seems feasible on paper, though, may not be quite so simple in practice ... That's why a new class of message-oriented middleware is appearing, designed precisely to iron out the uncertainties of remote networking. [Purpose of paper]
The world of ubiquitous computing is getting tantalizingly near ... Not so long ago only the largest corporations could afford to have global WANs ... Nowadays even the smallest companies can install remote networking software... [Contextualization]

Product description: Dial-up communications is still something of a black art ... In addition, dial-up and wireless communications are far less reliable than the office LAN ... Web browser/ Web server software simply does not provide the scalability, availability, reliability, security, or high performance that such services need... [Problem]
Messaging middleware attempts to tackle all these issues ... Media Transfer, from the French firm Telelogos, automates and secures file transfers between remote machines and a central server ... These products cover the key areas of asset and configuration management, software distribution, diagnostics and fault recovery... [Solution]

Conclusion/evaluation: Messaging middleware offers solutions to weaknesses in communication software, but it may not have a long-term future ... As more operators introduce services like AWDE, the cellular network will become a more useful means of data communications...

Reference to further information: Information Transport Associates Annapolis, MD, U.S. +410-280-3550
fax: +410-280-3553 info@itacorp.com
http://www.itacorp.com/...

Now each section (as outlined above) will be studied separately.

4.1.1. *Name of section*

Most computing science popular magazines are divided into sections. This facilitates the reading, since buyers know the contents of the magazine (whatever the products or the problems reviewed) in advance. Knowledge of what the article will deal with (whether the author will suggest solutions to a problem or will review a new product) avoids the reader the tiring task of browsing in order to guess the purpose of the text. Thus, usual buyers know that the article contained in the *First Looks* section will offer a description of the features of a new product; and the text under *Internet User Solutions* will provide them with very concrete suggestions regarding Web surfing.

The name of the section is usually inserted at the top of the page, in a different and —normally— bigger font (with respect to that used for the body text), and sometimes accompanied by a related picture (for example, a sort of floppy-disk drive, drawn in an unrealistic manner, precedes the words *PC Tech Solutions* —the name of the section— in a black box heading the article *Palm Computing: So Much from So Little*).

4.1.2. *Title*

As noted in the schematic structure above, the name of a product (whether a commercial brand, e.g. *Windows, Mac*; or a generic name referring to a computer-related machine or device, e.g. *modem*, *keyboard*) usually appears in the title, often accompanied by some type of comment. This may be a value judgment, like in *Palm Computing: So Much from So Little*, where the author emphasizes and summarizes the —still unknown for the reader— advantages of the product: *Palm Computing*. Or a merely informative phrase (with no biased comment), like in *Internet Apps in Win 98/NT 4*, which clearly informs the reader about the content of the article (a listing of Internet protocols and applications in a particular operating system). Other titles containing the name of a product are: *The Virtual*

Design Team, Little Languages with Lex, Yacc, and MFC, Compaq Takes Aim at Direct Vendors, and *C++ Containers Are Not Their Elements*.

The presence of product names in titles is due to the fact that, as previously said, popular computing science texts basically aim at presenting new information about a specific product. That's why when the title of one of the papers studied does not include the name of a product, it anyway contains pieces of information that refer directly to the type of product which is being reviewed. For instance, in *Get the Message*, the noun *message* regards the world of communication and remote networking, which are key topics of the article; besides, the word *message* comes up recursively throughout the text.

4.1.2.1. *Pre- and post-title headlines*

The authors of the papers under study often include pre- and post-title headlines that supplement the information given in the title about the contents of the article. Normally in italics and in a bigger font, these headlines are used with a double purpose: firstly, they introduce, but also summarize, the information offered in the article in order to provide the reader with a brief account of what the text is about; and, secondly, they highlight certain features of the body text that could persuade the consumer of these magazines to read the paper. This can be observed in the following examples:

> Now you can order Compaq's new Prosignia line of desktops, notebooks, and servers direct, over the phone or from the vendor's Web site. (Compaq Takes Aim at Direct Vendors, p. 56).
> A Computational simulation model of project organization that is usable and predictive for routine, project-oriented design tasks. (The Virtual Design Team, p. 84)
> Slide provides an IDE for little languages (Little Languages with Lex, Yacc, and MFC, p. 28)

In the first example (a post-title headline), the author's use of advertising/commercial style tries mainly to attract the attention of potential readers. The second one (a pre-title headline) combines the two functions: it explains what the product in the title (*The Virtual Design Team*) is and what it is for; and, in doing so, employs adjectives that could be considered as slightly positive in meaning (thus appealing readers): *usable* and *predictive*. Finally, the third example (a post-title headline), the most concise of the three, consists only of a merely informative sentence.

4.1.3. *Author*

The name/s of the author/s of the paper usually come/s after the title and before the post-title headline (if there is one), accompanied by the preposition *by*. It is not always so, however, since the author's name may sometimes be preceded by the post-title headline; or, in other cases, may be given as a marginal note and functioning as a caption to the photograph of the author (e.g. *C++ Report* articles).

Readers of this type of computing science papers are also likely to find details about the author's CV somewhere in the article. This information may appear on the front page (in a margin or at the end of the first column of the text, always somehow separated from the main body text) or at the end of the article. With respect to the length of this sub-section, a great deal of information (including present post and e-mail address) may be given, like in these examples:

> Andrew Koening is a Principal Research Staff Member at AT&T Research. He is the C++ columnist for JOOP, coauthor of Ruminations on C++, and author of C Traps and Pitfalls, and project editor for the ISO/ANSI C++ Committee. He can be contacted at ark@research.att.com. (*C++ Containers Are Not Their Elements*, p. 13)
>
> Jason is a programmer at Maxis, where he develops simulators and simulation languages. He can be contacted at jshankel@maxis.com. (*Little Languages with Lex, Yacc, and MFC*, p. 28)

But magazines do not always include that amount of data about their contributors. Sometimes they only specify information relevant to the magazine itself, like in the following example, where no reference to outside posts or contact addresses are included:

> Neil Randall is a contributing editor of PC Magazine. (*Palm Computing: So Much from So Little*, p. 268)

4.1.4.1. *Introduction*

4.1.4.1.1. *Purpose of paper*

The introductory section of the computing science popular paper normally contains a sub-section where the author presents the purpose of the paper, as well as another subsection which situates (in time, place, and with respect to other computing elements) the main topic of the article.

The aim of the paper is not always explicitly expressed, although sometimes, like in *Little Languages with Lex, Yacc, and MFC* and *Compaq Takes Aim at Direct Vendors*, the purpose of the paper is so clearly formulated that it even resembles the style of an academic paper (mainly the first example):

> In this article, I will describe how to use lex, yacc, and MFC to create integrated Win32 development environments for little languages. More specifically, I'll develop a multidocument-interface application called «Slide» (short for «Small Language Integrated Development Environment») and integrate it with lex and yacc. (*Little Languages with Lex, Yacc, and MFC*, p. 28)
>
> So what follows is a face-off between these PCs and what will be their major rivals, the Dell Inspiron notebooks and Dimension desktops. (*Compaq Takes Aim at Direct* Vendors, p. 56)

Expressions like *In this article, I will describe...; More specifically, I'll develop...*; or *So what follows is...*, clearly indicate what

the author is doing (or, at least, intends to do) in the following pages.

But the reader does not always avail of these markers expressing the aim of the article. Sometimes what the writer intends to achieve is expressed through a rhetorical question normally contained in the first paragraphs of the body text. It must be understood as a question which will be answered throughout the article, like in the following example, in which the lack of an immediate proper answer indicates the reader that the question is rhetorical and that it relates directly to the purpose of the paper.

> How does so small a package accomplish so much? (*Palm Computing: So Much from So Little*, p. 263)

Sometimes, though, there is not even a rhetorical question that can help the reader find out the purpose of the paper. When such markers do not exist, the reader will still be able to combine loose pieces of information throughout the introductory section in order to make up what the writer intends to do in the article. Most times this information can be also inferred from the contents of other sections, such as the title, the pre- and post-title headlines, the contextualization, and even the name of the author and the section of the magazine. The article *C++ Containers Are Not Their Elements*, for instance, does not explicitly expresses the purpose of the paper in the introduction. Notwithstanding, the combination of different sentences from the contextualization, as well as the name of the section where the article is inserted (*Traps and Pitfalls*) help the reader determine this information. The contextualizing parts referred to above are:

> In a previous article1 [this superscript number makes reference to a final note where details of this previous article are given], I pointed out a commom mistake that programmers make while learning how to use the C++ standard library. (*C++ Containers Are Not Their Elements*, p. 13)

> The naive use of copy fails because iterators make available only a limited subset of the possible container operations. These limitations constrain not only uses of iterators, but also uses of algorithms. (*C++ Containers Are Not Their Elements*, p. 13)

Bearing in mind that these are the initial and final sentences of the introductory section of this article, usual readers of this kind of magazines will not take long to infer that the purpose of the paper is to give suggestions in order to avoid some errors related to the programming of the C++ standard library.

4.1.4.1.2. *Contextualization*

The contextualization, although *a priori* an integral part of the introductory section, may sometimes appear inserted in several other sections of the paper, since the different stages in the description of a product, for example, may have their own contextualizing sub-sections. However, here we are referring only to the general contextualization of the topic to be dealt with in the article (a simple description of a product, a comparison of two products or a problem-solution type of text[1]).

The contextualization section normally offers brief background information about the development of the product to be reviewed in the article, or some comment on the current situation of a computer-related topic and how it has developed since it was first considered. The frequent use of time and place expressions (in order to physically situate what is being described) marks this section, as these examples show:

> The 3Com Palm III personal digital assistant (PDA) and its older siblings have become increasingly popular since the system was introduced thanks to their simplicity, functionality, and ease of use. (*Palm Computing: So Much from So Little*, p. 263)

---

[1] For a detailed description of this type of texts, see Hoey (1994:26-45).

> As with all computing machinery, the Palm relies on an operating system for its functionality. Like a networked PC, which includes both an OS and a means of linking to the network operating system, the Palm includes both the OS (currently Palm OS, Version 3.0) and software that lets the device link with a PC or a Mac. (*Palm Computing: So Much from So Little*, p. 263)
>
> What better place to do battle with your rivals than on their turf? This is the approach that Compaq Computer Corp. is taking with the mid-November introduction of its Prosignia line of desktops, notebooks, and servers for small businesses. (*Compaq Takes Aim at Direct Vendors*, p. 56)

In the first example, the author briefly introduces the product which is going to be reviewed and gives account of its successful development. In the following one, the product referred to in the title is presented and defined with respect to other computing devices, so it can be also considered a way of contextualizing. Finally, in the third example, the date of the introduction of the new *Prosignia* line marks the beginning of a fictional «battle», which will be represented in the article by means of a comparison of these new *Prosignia* products and their major competitors.

4.1.4.2. *Product description*

Although most computing science popular papers follow a same generic pattern or schematic structure, not all of them are devoted to the same type of information. Some of them merely deal with the presentation of the features of a new product; some compare two rival products (or an upgraded product with its older version); and others present a problem-solution text type, where a more or less general computing problem is introduced in the first place, and then a suggestion is given in order to solve this problem. We have used the generic phrase «Product description» for this section as all of the articles studied, although slightly differently organized according to which of the three types of information they contain, have the analysis of one or more products as a defining feature.

Now let us observe how the different types of information determine the way the text is organized. It must be taken into account that authors very frequently combine the three types of information, thus producing mixed texts and putting together some of the characteristics explained below.

Presentation of the features of a product

Although it is very unlikely to find a text where the features of a product are presented and no reference is made to other products or to generic computer-related problems, there exist some articles where the focus of the paper is mainly on the detailed description of some particular machine or device. These texts are normally divided into sub-sections (preceded by headings which determine what information is given next). For instance, in the article *Palm Computing: So Much from So Little*, details about *memory and data storage, power modes, graffiti* and *hotsync and conduits* (all features of the new *Palm* personal digital assistant) are presented consecutively, providing the reader with a general picture of the machine. Some excerpts from this article show the recursive use of typical linguistic features:

> The device's ROM holds the operating system plus the applications that ship with the unit; RAM is used for any applications you install or develop yourself, user data for all applications, and any other user-configurable data. Because the RAM is powered by the Palm's batteries, data is safe when the Palm enters low-power modes (see «Power Modes» below). You will lose data in RAM if the Palm is without power for over a minute, however, so it's important to transfer your data to your PC before losing battery power (as when you change batteries). (p. 263)
>
> Switching from one application to another is instantaneous, because primary memory simply closes the first and launches the second, with the modified data also stored in primary memory. In other words, data never leaves the nonvolatile primary memory in which it is stored, so there's no need for an external storage device in which to preserve it. (p. 264)

Present simple tense verbs, conditional constructions (expressing a cause-effect relation), series of alternative possibilities (linked by means of commas and conjunctions), and expressions intended to facilitate and clarify the reading (*so it's important to, in other words*) are some of the linguistic features employed by the author in order to present the product reviewed as clearly as possible.

Comparison of two products

Some sections in computing science popular magazines are exclusively devoted to the description of new products (e.g. *First Looks* in *PC Magazine*). The authors of the articles included in these sections normally compare the new products with other releases of the same type, or with older versions of the same machines. The following example presents the linguistic features used recursively by these writers:

> Comparing one vendor's non-corporate-focused desktops with another's —strictly on the basis of hardware— is a relatively easy task, as most use the same core components. And that's what we found when comparing the Prosignia Desktop 330 with a Dell Dimension XPS R450. (*Compaq Takes Aim at Direct Vendors*, p. 56)
>
> Subtle differences do exist. Whereas the Compaq PC uses a 16.8GB hard disk, the Dell can be ordered with a new 17.2GB disk. Both have speakers in their standard configurations, but the Dimension's speakers (an Altec lansing ACS295 system with subwoofer) have superior sound. You can, however, order the Prosignia with Altec Lansing ACS48 speakers and subwoofer for an additional $85. Both machines are mini-towers, but the Prosignia can be reconfigured as a desktop; the Dimension, on the other hand, provides easier internal access via a single thumbscrew. (*Compaq Takes Aim at Direct Vendors*, p. 56)

The use of comparative adjectives (*superior, easier*) and expressions (*comparing ... on the basis of, when comparing*), nouns denoting «difference» (*differences*), and the «alternating» style of the text (similar features of the two products reviewed are alternately descri-

bed with the help of linguistic markers, such as *whereas, however, on the other hand*) help the reader notice that the author is comparing two different products.

Problem-solution text type

Computer-related problems are constantly referred to in the articles under study. Most times they appear in order to highlight the advantages of a product (as long as certain features of the product can handle those problems), and are not given an in-depth treatment. On some occasions, however, the body text is organized as a two-stage structure, where the first stage is the presentation of a more or less general problem, and the second one, the proposal of available solutions. In the article *Get the Message*, for instance, approximately a quarter of the body text is devoted to explaining how much the difficulties of remote communication through networking software can affect the everyday transactions of all kinds of a company. An example of this problem stage can be observed in the following excerpts:

> Dial-up communications is still something of a black art, involving start-up scripts, Winsock settings, and other unfriendly stuff that can make life hell for a salesman in a hurry in a motel room. (p. 7)
>
> «The Internet infrastructure is far from sufficient for supporting business-critical transactions,» says Rosemary Rock-Evans, an associate at the U.K. consultancy Ovum. «Web browser/Web server software simply does not provide the scalability, availability, reliability, security, or high performance that such services need.» (p. 7)
>
> The weakness of such solutions usually is the software. Any solution that relies on ordinary Windows communications products is likely to be too difficult and too unreliable for most salespeople. Then there's the problem of navigation once a user is logged on, finding the right applications and databases and issuing the appropriate queries. (pp. 7-8)

Certain linguistic features, such as the use of negative-meaning words (adjectives: *black, unfriendly, difficult, unreliable*; nouns: *hell,*

weakness, problem), phrases (*far from sufficient*) and sentences (*Web browser/Web server software simply does not provide the scalability, availability, reliability, security, or high performance that such services need*), as well as the style, focused mainly on the practical consequences of the problem, are to be considered as defining characteristics of this stage.

After the problem has been explained, one or more available solutions are presented. In the article used as example (*Get the Message*), the author offers several products (divided by headings) and evaluates the way each of these tackle the communications problem (the characteristics of this part of the body text are as shown in *Presentation of the features of a product* above). However, hints of a solution are already included throughout the problem stage, as can be observed in the following excerpts:

> That's why a new class of message-oriented middleware is appearing, designed precisely to iron out the uncertainties of remote networking. (p. 7)
> This approach is particularly helpful when you're using the World Wide Web. (p. 7)
> . The result is much quicker queries, which can be very important when you're sitting face-to-face with a customer. (p. 8)

The great variety of positive-meaning linguistic elements (words, phrases and whole sentences) and the highlighting of advantages of a new group of products, contrast with the negative context (as explained above) where these excerpts are inserted.

4.1.4.3. *Conclusion/Evaluation*

Authors of this type of articles usually give a final evaluation of the products they have described in the text. This sort of conclusion, which does not always appear in such a manner, normally emphasises the advantages of what has been reviewed, including the aut-

hor's own value judgments and indirectly appealing the reader to buy the product, as the following examples show:

> As more GSM operators introduce services like AWDE, the cellular network will become a more useful means of data communications. Liberating the sales force from office and hotel telephones is useful in the industrialized world. In countries where the public telephone infrastructure is sparse or nonexistent, it's a necessity. (*Get the Message*, p. 11)
>
> As you can see, the Palm OS ensures that this little PDA manages data properly. It allows fast, easy, and usable input, and it synchronizes efficiently with desktop software running on the PC. No wonder the device has won the hearts and minds of so many. (*Palm Computing: So Much from So Little*, p. 268)

This —in some cases exaggerating— final account of advantages may, however, not appear at the end of the article (as in the examples above). Sometimes it could even seem that there is no conclusion at all, since no linguistic markers (such as lexical items denoting summarizing or global evaluation, phrases like *as you can see*, or the change from a merely informative style to a more opinion-oriented one) are to be found in the papers. Readers can then resort to the title or pre— and post-title headlines (which could include some kind of evaluation); some figures' captions (where a summary of the article is frequently offered); or the in-text information boxes (which are often used by authors to give their own views far from the technical data of the body text).

4.1.4.4. *Reference to further information*

Unlike research academic papers, computing science popular articles rarely include bibliographical information. The type of references that authors of these texts most frequently provide the reader with is practical details, such as telephone numbers and Web site addresses related to the product or products reviewed and the companies that produce them. This information, usually contained in

marginal notes and in in-text information boxes (not necessarily divided from the rest of the body text), can be presented in a schematic way, as shown in the following examples:

> **Compaq Prosignia Desktop 330.** $2,239 direct. Compaq Computer Corp., Houston; 800-888-9909; www.compaq.com/smb. (*Compaq Takes Aim at Direct Vendors*, p. 56)

Name of product; price; name of company, its telephone number and its Web address, are the practical data offered to the readers interested in further contact and information.

The following linear diagram represents the schematic structure which we have described stage by stage above (under Eggins 1994: 40):

(Name of section of magazine) ^ (Pre-title headline) ^ Title ^ (Post-title headline) ^ Name of author/s * Purpose of paper * Contextualization ^ Product description ^ (Conclusion) * (Reference to further information)

Key: ^ indicates fixed order, * unordered stage, ( ) optional stage

5. Linguistic features in texts

After the description of the structure of the texts, we will pay attention to certain linguistic features which co-occurred in all of them. Those features are the use of the present simple tense, comparatives and superlatives, expressions of reason, purpose and result, causative verbs like *allow, let* and *enable*, the modal verb *can* and the construction *by + -ing*.

5.1. *Use of the present simple tense*

As in this sort of texts there is a description of a product, it is usual to find a great deal of verbs in the present simple. This tense is used for explanations and descriptions on how something works,

what it can do and what it is for. It frequently appears in explanations with expressions like *for example* and *for instance*:

> The remote run command works via «robot» programs that can control existing applications by feeding them keystrokes... (*Get the Message*, p. 8)
>
> When the stylus touches the Graffiti display area, the event manager calls the pen manager and informs it of the pixel location. (*Palm Computing: So Much From So Little*, p. 268).
>
> Again, as soon as the Palm detects user input it comes out of sleep mode and goes directly into running mode. (*Palm Computing: So Much From So Little*, p. 268).

The use of the present simple tense in the first example (*works*) obviously aims at explaining how something (*the remote run command*) works; and, in the second and third examples, at describing the process of what happens when a certain action is carried out or something takes place (*touches, calls, informs, detects, comes, goes*). The actions are expressed by means of this verb tense since they always occur.

5.2. *Expressions of reason, purpose and result*

Due to the informative nature of the sort of texts we are dealing with and the contents they present, the relations of reason (cause and effect, reason and consequence), as well as expressions of purpose and result appear very frequently.

5.2.1. *Reason relationships*

Cause and effect

Cause may be indicated by an adverbial *because*-clause, by a prepositional phrase using the expressions *because of, from, out of,* by some verbal constructions like *lead to, result in, give rise to* or by some causative verbs. However there are other ways of expressing this relation as we will show in the third example below:

> Because the RAM is powered by the Palm's batteries, data is safe when the Palm enters low-power modes. (*Palm Computing: So Much from So Little*, p. 263)
>
> Because of the way in which data is stored, and because of the user's need to run a PDA instantly on demand, the power system is crucial. (*Palm Computing: So Much from So Little*, p. 264)
>
> Given that e-mail and the Web are two major Internet activities and that FTP remains a download protocol of choice for many sites, IE and OE take the brunt of most user's work. (*Internet Apps in Win 98/NT 4*, p. 241)

In the first example the cause for the data being safe is that the RAM is powered by the batteries, and this is expressed by an adverbial *because*-clause.

The prepositional phrase using the expression *because of*, which appears in the second example, states that the causes for the power system being crucial are the way in which data is stored and the user's need to run a PDA instantly on demand.

In the last example, the expression *given that* may be replaced by the conjunction *because*.

Reason and consequence

Because and *because of* can express reason as well as cause. Cause and reason are similar and thus confusing notions as both answer the same question: why? But we can see a difference between them in that reason concerns the speaker's inference of a connection, and cause expresses the perception of an inherent objective connection.

Reason can also be expressed by *as*-clauses and *since*-clauses. We can say that the main clause indicates the consequence of the reason clause. There are linking adverbials of cause or reason meaning *because of that* or *for that reason*: *therefore, thus, so, accordingly, consequently*. Examples are:

> The best way to access it is through the MS-DOS prompt, because if you launch it directly from Start/Run, the ping window will shut down before you have time to read the information. (*Internet Apps in Win 98/NT4*, p. 242)
>
> There is a driver under Windows 98 to completely virtualize the Sound Blaster function under Windows-DOS box, thus allowing comprehensive DOS game support. (*Your PC Never Sounded So Good*, p. 14)
>
> ... users don't care about trees and directed graphs and therefore never notice their absence. (*Matlab as a Scripting Language*, p. 99)

In the first example, the reason that the best way to access it is through the MS-DOS prompt is that the ping window will shut down before you have time to read the information. In this case the subordinate clause is the consequence of the main clause.

The linking adverbs *thus* and *therefore* express the consequence of the main clauses in the second and third examples, respectively.

5.2.2. *Purpose*

The purpose of an action is described by an adverbial of purpose, which is usually a *to*-infinitive clause, but it may also be a finite verb clause beginning *so that* or *so*. However, the construction *for* +-*ing* also expresses purpose, as it refers to the function of something or to what something is for:

> To get a list of commands to use once and FTP connection has been established, type help at the FTP prompt. (*Internet Apps in Win 98/NT4*, p. 242)
>
> The ping tool has one simple purpose: to let you know whether a remote machine is accessible. (*Internet Apps in Win 98/NT4*, p. 242)
>
> The first is to specify a Matlab structure for storing the data defining instances of that class. (*Matlab as a Scripting Language*, p. 97)

In the first example, the *to*-infinitive clause *(to get a list)* is the purpose of typing *help* at the FTP prompt. In the second example the purpose is explicitly expressed by the sentence *The ping tool*

has one simple purpose:, which is followed, again, by a *to*-infinitive clause *(to let you know...)*.

In the third example, the purpose of a Matlab structure is that of storing the data defining instances of that class, which is expressed by the construction for + -ing. As we stated above, it expresses what that certain structure is for.

5.2.3. *Result*

It can be expressed by a clause beginning with *so (that)*. Although result clauses overlap with purpose clauses, there is an important and clear difference: while in the purpose clause the result is yet to be achieved, in the result clause, it is already achieved.

In the texts selected there are other ways of expressing result as we will observe in the examples below:

> ...Windows users today are rarely comfortable with DOS, let alone Unix, ... As a result, a huge number of freeware and shareware FTP utilities have been developed for Windows... (*Internet Apps in Win 98/NT4*, p. 242)
>
> Most finger information is now available on Web pages or through institutional directories, so finger is less used all the time. (*Internet Apps in Win 98/NT4*, p. 242)
>
> The result is speed. (*Palm Computing: So Much From So Little*, p. 263)
>
> This resulted in a language that, ..., fits the needs of physical scientists and engineers very well. (*Matlab as a Scripting Language*, p. 94)

In the first, third, and fourth examples, the result relationship is explicit with the expressions *as a result, the result is* and *this resulted in.* The second example shows this relation through the use of the subordinator *so,* by which it is stated that the fact that finger is less used all the time is the result of the fact that most finger information is now available on Web pages or through institutional directories.

5.3. *Causative verbs:* allow, let, enable

In this type of discourse we can find examples of very usual causative verbs, such as *allow, let* and *enable*. The infinitive which normally follows them expresses the resultant action. The cause is often the subject of the sentence:

> These servers are HTTP daemons, and they allow Web clients to retrieve data. (*Internet Apps in Win 98/NT4*, p. 241)
> ... the Palm includes both the OS and software that lets the device link with a PC or a Mac. (*Palm Computing: So Much from So Little*, p. 263)
> It also enables the user to defer connection until favored times... (*Get the Message*, p. 7)

In the first example, what causes clients to retrieve data are the servers, which, replaced by *they,* act as the subject. The same happens in the second and third examples with the use of *let* and *enable*, where *that* (the software) and *it* are respectively the causes by which the device links with a PC or a Mac and the user defers connection until favored times.

5.4. *Comparative and superlative adjectives and adverbs*

As these texts are normally about a new computer product or an improved version of a certain product, it is very common to find comparisons between them and others of the same kind or the latest versions in order to show the advantages and the development and changes which have taken place. Taking into account the fact that in most of these texts it is possible to find the references to further information and the prices of such products, the advertising function is obvious and consequently the need to highlight the qualities and advantages. That is the reason why it is usual to find comparatives and superlatives in this sort of texts. Instances of this are:

> It occurs a much wider range of frequencies than either the Sony or the Ten-Tec product. (*Boot Up, Turn On, Tune In*, p. 56)

> Programming languages are as prone to «feature bloat» as everything else in computing. (*Matlab as a Scripting Language*, p. 98)
>
> The largest change, though not the most significant was the multidimensional array. (*Matlab as a Scripting Language*, p. 99)

In the first example, the adjective *wider* expresses the comparison relationship between the *Sony* or the *Ten-Tec* product and the other product which has been mentioned before through the pronoun *it* meaning PCR 1000.

When expressing similarities between products, as in the second example, the equal comparison is used: *as prone... as.*

In the third example, the superlatives *largest* and *most significant* are used to compare more than two objects. Of all the changes which took place, the largest was the multidimensional array.

5.5. *Use of the modal verb* can

The frequent use of the modal verb *can* expressing «possibility» is due to the fact that the writer of the text tells the reader about the possibilities offered by the computer product and its achievements. Also, *can* can be generally changed by the string *it is possible* followed by an infinitive clause:

> The scripts can send or receive files, delete or rename files, run a command on the remote computer, and synchronize the time and date between remote machines. (*Get the message*, p. 8)
>
> Users can customize, among other things, window style, OLE support, control files, ... (*Little Languages with Lex, Yacc, and MFC*, p. 28)

The first example could be here paraphrased by *It is possible for the scripts to send or receive files, delete;* and the second example by *users have the possibility to customize window style, ...*

5.6. *Construction* by + -ing

This construction is representative of this type of texts and it is used to explain how and by means of what something works and how the user of the computer product can achieve his aims. Phrases of mean answer the question *How...? By* can express the meaning *by means of*:

> By specifying the user name anonymous and using a generic password, you can access directories that the operators of a remote server have set aside specifically for publicly available files, usually in a directory named Pub. (*Internet Apps in Win 98/NT4*, p. 241).

The sentence above answers the question *How can I access directories that the operators of a remote server have set aside...?* This common construction in the type of texts we are dealing with expresses a similar meaning to reason clauses. The sentence would mean the same as *To access directories that the operators of a remote server have set aside specifically for publicly available files, usually in a directory named Pub, specify the user name «anonymous» and use a generic password.*

Conclusion

This paper has highlighted the internal structure of computing science papers which are included in computing magazines. The study has revealed a certain schematic structural pattern which proves specific of this sort of texts. This schematic pattern is organised into different units of information, called *stages*, which conform the overall structure of the text. In addition, each stage contains a specific use of language, such as the use of causative verbs and comparative expressions, which helps the reader recognise the type of text he is reading. We have also studied the use of typographical elements used in this type of texts, mainly because they greatly contribute to the generic structure of computing science magazine texts.

The above study stands as a first step towards the analysis of computing science magazine articles as a specific text-type, and hence, there are many aspects which have not yet been covered within the framework of this paper. Future research in this area must include information at a more pragmatic level, such as the use of language intending direct addressing to the reader, or even the presence of what we have called here in-text information boxes, which we felt should be studied on their own.

WORKS CITED

Primary Sources
Freed, L. 1999. «Boot Up, Turn On, Tune In» in *PC Magazine,* January 5: 56-57.
Hummel, R. L. and **D. Andrews**. 1998. «Notebook Screens to Get Bigger and Better» in *Byte,* July: 20-21.
Kao, S. 1998. «Your PC Never Sounded So Good» in *Byte,* July: 13-19.
Koenig, A. 1998. «C++ Containers Are Not Their Elements» in *C++ Report,* November-December: 13-14.
Kunz, J. C. et al. 1998. «The Virtual Design Team» in *Communications of the ACM,* Vol. 41, No. 11: 84-91.
Martin, R. C. 1998. «UML Use-Case Diagrams» in *C++ Report,* November-December: 49-51.
Moore, R. et al. 1998. «Data-Intensive Computing and Digital Libraries» in *Communications of the ACM,* Vol. 41, No. 11: 56-62.
Pountain, D. 1998. «Get the Message» in *Byte,* July: 7-11.
Randall, N. 1999. «Internet Apps in Win 98/NT 4» in *PC Magazine,* January 5: 241-242.
Randall, N. 1999. «Palm Computing: So Much from So Little» in *PC Magazine,* January 5: 263-268.
Shankel, J. 1999. «Little Languages with Lex, Yacc, and MFC» in *Dr. Dobb's Journal,* January: 28-33.
Tabibian, O. R. 1999. «Sun Ups the Ante» in *PC Magazine,* January 19: 75.
Venezia, C. 1999. «Compaq Takes Aim at Direct Vendors» in *PC Magazine,* January 5: 56-7.
Webb, P. and **G. V. Wilson**. 1999. «Matlab as a Scripting Language» in *Dr. Dobb's Journal,* January: 94-99.

Secondary Sources

Eggins, S. 1994. *An Introduction to Systemic Functional Linguistics.* London: Pinter.

Hoey, M. 1994. «Signalling in Discourse: a Functional Analysis of a Common Discourse Pattern in Written and Spoken English,» in Coulthard, M. (ed) *Advances in Written Text Analysis.* London: Routledge, 26-45.

Leech, G. and **J. Svartvik**. 1975. *A Communicative Grammar of English.* Longman.

Quirk, R. et al. 1985. *A comprehensive Grammar of the English Language.* Longman.