

SDCC Progetto B1: Multicast totalmente e causalmente ordinato in Go

ALESSANDRO CHILLOTTI

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:

Alessandro Chillotti. 2018. SDCC Progetto B1: Multicast totalmente e causalmente ordinato in Go. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 6 pages. <https://doi.org/10.1145/1122445.1122456>

1 TRACCIA DEL PROGETTO

Lo scopo del progetto è realizzare nel linguaggio di programmazione Go un'applicazione distribuita che implementi gli algoritmi di multicast totalmente ordinato e causalmente ordinato.

L'applicazione deve soddisfare i requisiti elencati di seguito.

- Un servizio di registrazione dei processi che partecipano al gruppo di comunicazione multicast. Si assuma che la membership sia statica durante l'esecuzione dell'applicazione, quindi non vi sono processi che si aggiungono al gruppo od escono dal gruppo durante la comunicazione.
- Il supporto dei seguenti algoritmi di multicast:
 - (1) multicast totalmente ordinato implementato in modo centralizzato tramite un sequencer;
 - (2) multicast totalmente ordinato implementato in modo decentralizzato tramite l'uso di clock logici scalari;
 - (3) multicast causalmente ordinato implementato in modo decentralizzato tramite l'uso di clock logici vettoriali.

Si richiede di testare il funzionamento degli algoritmi implementati nel caso in cui vi è un solo processo che invia il messaggio di multicast e nel caso in cui molteplici processi contemporaneamente inviano un messaggio di multicast; tali test devono essere forniti nella consegna del progetto.

Per il debugging, si raccomanda di implementare un flag di tipo verbose, che permette di stampare informazioni di logging con i dettagli dei messaggi inviati e ricevuti. Inoltre, per effettuare il testing in condizioni di maggiore stress, si consiglia di includere nell'invio dei messaggi un parametro delay, che permette di specificare un ritardo, generato in modo random in un intervallo predefinito.

Si progetti l'applicazione ponendo particolare cura al soddisfacimento dei requisiti sopra elencati. Si richiede inoltre che gli eventuali parametri relativi all'applicazione e al suo deployment siano configurabili.

Author's address: Alessandro Chillotti, alessandro.chillotti@outlook.it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/8-ART111 \$15.00

<https://doi.org/10.1145/1122445.1122456>

2 ASSUNZIONI

Le assunzioni che sono state fatte per realizzare questo applicativo sono:

- La comunicazione è affidabile, ovvero non è possibile che ci sia la presenza di messaggi persi;
- La comunicazione è *FIFO* ordered, ovvero se un processo p_i invia molteplici messaggi al processo p_j , quest'ultimo riceve i messaggi nello stesso identico ordine in cui p_i li ha inviati;
- È stato assunto un ritardo massimo nell'invio del messaggio pari a 3 secondi perché ritenuto realistico ed adatto allo scopo in questione. Questo ritardo non è altro che uno sleep del processo mittente, quindi viene generato un numero pseudo-randomico fra 0 e 3 al momento dell'inoltro del messaggio.

3 SCELTE PROGETTUALI

In questa sezione sono presentate e motivate le scelte progettuali effettuate in fase di sviluppo dell'applicativo.

3.1 Consegna di un messaggio di multicast

Tipicamente gli algoritmi di multicast vengono eseguiti al livello del middleware e quindi un importante aspetto da considerare nell'implementazione degli algoritmi è sicuramente la consegna a livello applicativo. Infatti, si ha una sostanziale differenza fra i termini *ricezione* di un messaggio e *consegna* di un messaggio. In particolare:

- La *ricezione* riguarda la fase in cui il messaggio arriva al nodo desiderato.
- La *consegna* riguarda la fase in cui il messaggio viene consegnato all'applicazione al livello sovrastante.

Poiché gli algoritmi sono già eseguiti a livello applicativo, è stato scelto di simulare la consegna di un messaggio attraverso il salvataggio su un file.

3.1.1 Struttura del file di consegna. Il file di consegna è visto come una tabella con i seguenti campi:

- *Id*: questo è un campo speciale perché rappresenta l'oggetto chiave con cui l'algoritmo lavora per mantenere l'ordinamento desiderato. In particolare, varia in base all'algoritmo selezionato:
 - Algoritmo 1: questo campo corrisponde all'identificativo assegnato dal *sequencer*.
 - Algoritmo 2: questo campo corrisponde al valore del clock logico scalare del peer mittente, nel momento in cui invia il messaggio.
 - Algoritmo 3: questo campo corrisponde al valore del clock logico vettoriale del peer mittente, nel momento in cui invia il messaggio.
- *Timestamp*: questo campo rappresenta l'istante temporale in cui quel messaggio è stato inviato dal peer mittente.

- *Username*: questo campo rappresenta lo username con cui un peer si identifica¹.
- *Messaggio*: questo campo rappresenta il contenuto del pacchetto effettivo.

Con lo scopo di facilitare la comprensione della struttura del file di log, viene mostrato un esempio che rappresenta lo snapshot, ad uno specifico istante temporale, del file di log nel caso in cui si adottando l'algoritmo 1.

Table 1. Esempio del file di consegna (Algoritmo 1)

<i>Id</i>	<i>Timestamp</i>	<i>Username</i>	<i>Messaggio</i>
1	12:59:32	peer_1	messaggio 1
2	13:02:17	peer_2	messaggio 2
3	13:05:27	peer_3	messaggio 3
4	13:08:17	peer_2	messaggio 4

3.2 Architetture

Per lo sviluppo degli algoritmi si è scelto di adottare due architetture differenti, in modo tale da poter scegliere di averne una più adatta in base all'algoritmo richiesto.

3.2.1 Scelte generali. Come descritto in precedenza, la consegna al livello applicativo è stata simulata con il salvataggio del contenuto del messaggio, più relativi metadati, in un file.

Il file risiede all'interno di volume *Docker* per ogni peer appartenente al gruppo multicast.

3.2.2 Architettura per l'algoritmo 1. L'architettura ideata per la realizzazione dell'algoritmo 1 è la seguente².

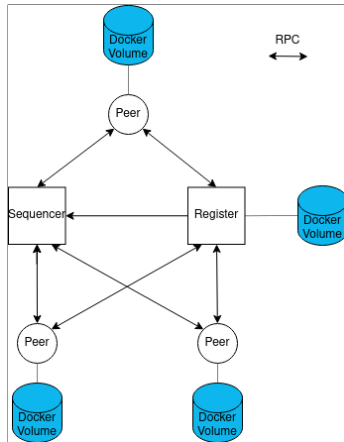


Fig. 1. Architettura algoritmo 1

Si può notare dalla figura che ad ogni peer è stato montato un volume *Docker*, nel quale è presente il file utilizzato per simulare la consegna del messaggio a livello applicativo.

¹Lo username è specificato in fase di registrazione.

²In figura è mostrata un'architettura con solamente 3 peer, ma questo è stato fatto solamente per semplificare l'architettura. Quindi, è possibile scegliere quale numero di peer far partecipare al gruppo multicast in fase di startup dell'applicazione.

Gli attori presenti nell'architettura sono:

- Il *peer* rappresenta un partecipante al gruppo di multicast. Ovviamente, per poter partecipare alla comunicazione si ha la necessità di eseguire una registrazione ad esso.
- Il *register* rappresenta il nodo che permette di:
 - Accettare le registrazioni dei peer finché non si raggiunge il numero di partecipanti stabilito al gruppo multicast.
 - Inviare la lista dei peer partecipanti al gruppo ad ogni peer registrato.
- Il *sequencer* è il nodo che implementa l'algoritmo, assegnando un identificativo ad ogni pacchetto che permette di avere un ordinamento totale.

3.2.3 Architettura per gli algoritmi 2 e 3. L'architettura ideata per la realizzazione degli algoritmi 2 e 3 è la seguente.

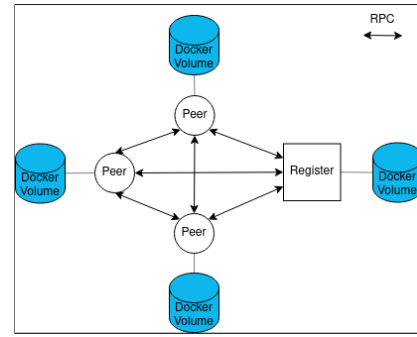


Fig. 2. Architettura algoritmi 2 e 3

Si può notare, come rispetto all'architettura precedente, non sia presente il *sequencer* poiché gli algoritmi 2 e 3 sono algoritmi realizzati in modo distribuito.

4 IMPLEMENTAZIONE

5 PIATTAFORMA SOFTWARE

6 TESTING DELL'APPLICAZIONE

Per verificare il corretto comportamento degli algoritmi implementati sono stati ideati dei test. In particolare, per ogni algoritmo:

- Un test riguarda l'invio del messaggio di multicast da parte di un solo peer.
- Un test riguarda l'invio del messaggio di multicast, anche in modo concorrente, da parte di più peer.

6.1 Test per l'algoritmo 1

6.1.1 Un solo sender. In questa tipologia di test un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO ordered*.

Il risultato atteso è che ogni peer consegna, nello stesso identico ordine, i messaggi ricevuti al livello applicativo.

6.1.2 Più sender. In questa tipologia di test ogni peer, in modo concorrente, invia un messaggio al *sequencer* e dopo aver inviato il primo messaggio, inoltra anche un secondo messaggio.

Il risultato atteso è che ogni peer consegna, nello stesso identico ordine, i messaggi ricevuti al livello applicativo.

6.2 Test per l'algoritmo 2

6.2.1 Un solo sender. In questa tipologia di test un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO* ordered.

Il risultato atteso è che nessun peer consegna messaggi a livello applicativo, in quanto non viene mai rispettata la condizione di consegna poiché è solamente un peer ad effettuare l'inoltro del messaggio in multicast.

6.2.2 Più sender. In questa tipologia di test ogni peer, in modo concorrente, invia un messaggio al *sequencer* e dopo aver inviato il primo messaggio, inoltra anche un secondo messaggio.

Il risultato atteso è che i primi N messaggi consegnati al livello applicativo da ciascun peer siano i medesimi, dove N è il numero minimo di messaggi consegnati dai peer.

6.3 Test per l'algoritmo 3

6.3.1 Un solo sender. In questa tipologia di test un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO* ordered.

Il risultato atteso è che ogni peer consegna, nello stesso identico ordine, i messaggi ricevuti al livello applicativo poiché, essendo un solo sender, non c'è relazione di causa-effetto fra i messaggi.

6.3.2 Più sender. In questa tipologia di test si è seguito uno schema ben preciso per avere un test valido dal punto di vista della verifica. Lo schema è rappresentato nella seguente figura.

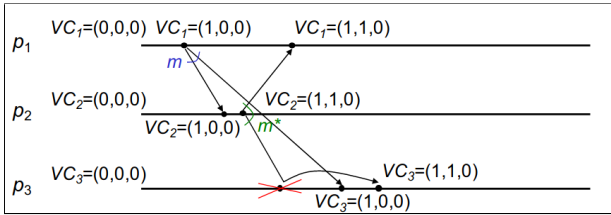


Fig. 3. Test per l'algoritmo 3

In particolare, lo scenario è il seguente:

- Il peer p_1 invia un messaggio con timestamp $(1, 0, 0)$ in multicast. Questo messaggio viene ricevuto dal peer p_2 , ma a causa di un ritardo non viene ricevuto subito dal peer p_3 .
- Il peer p_2 , a causa del messaggio ricevuto, inoltra un messaggio in multicast con timestamp $(1, 1, 0)$. Questo messaggio viene ricevuto da entrambi i peer senza ritardi.
- Il peer p_3 , al momento della ricezione del messaggio inviato da p_2 , deve posticipare la consegna del messaggio con timestamp $(1, 1, 0)$. Una volta ricevuto e consegnato il messaggio con timestamp $(1, 0, 0)$, può procedere con la consegna del messaggio inviato da p_2 .

Il risultato atteso è che tutti i peer consegnino i messaggi rispettando la relazione di causa-effetto. Quindi, in questo caso significa che ogni peer consegna, nello stesso ordine, i due messaggi al livello applicativo.

6.4 Template Styles

The primary parameter given to the “acmart” document class is the *template style* which corresponds to the kind of publication or SIG publishing the work. This parameter is enclosed in square brackets and is a part of the `documentclass` command:

```
\documentclass[STYLE]{acmart}
```

Journals use one of three template styles. All but three ACM journals use the `acmsmall` template style:

- `acmsmall`: The default journal template style.
- `acmlarge`: Used by JOCCH and TAP.
- `acmtog`: Used by TOG.

The majority of conference proceedings documentation will use the `acmconf` template style.

- `acmconf`: The default proceedings template style.
- `sigchi`: Used for SIGCHI conference articles.
- `sigchi-a`: Used for SIGCHI “Extended Abstract” articles.
- `sigplan`: Used for SIGPLAN conference articles.

6.5 Template Parameters

In addition to specifying the *template style* to be used in formatting your work, there are a number of *template parameters* which modify some part of the applied template style. A complete list of these parameters can be found in the *L^AT_EX User’s Guide*.

Frequently-used parameters, or combinations of parameters, include:

- `anonymous, review`: Suitable for a “double-blind” conference submission. Anonymizes the work and includes line numbers. Use with the `\acmSubmissionID` command to print the submission’s unique ID on each page of the work.
- `authorversion`: Produces a version of the work suitable for posting by the author.
- `screen`: Produces colored hyperlinks.

This document uses the following string as the first command in the source file:

```
\documentclass[acmtog]{acmart}
```

7 MODIFICATIONS

Modifying the template — including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions, and the use of the `\vspace` command to manually adjust the vertical spacing between elements of your work — is not allowed.

Your document will be returned to you for revision if modifications are discovered.

8 TYPEFACES

The “acmart” document class requires the use of the “Libertine” typeface family. Your $\text{T}_{\text{E}}\text{X}$ installation should include this set of packages. Please do not substitute other typefaces. The “lmodern” and “ltimes” packages should not be used, as they will override the built-in typeface families.

9 TITLE INFORMATION

The title of your work should use capital letters appropriately - <https://capitalizemytitle.com/> has useful rules for capitalization. Use

the `title` command to define the title of your work. If your work has a subtitle, define it with the `subtitle` command. Do not insert line breaks in your title.

If your title is lengthy, you must define a short version to be used in the page headers, to prevent overlapping text. The `title` command has a “short title” parameter:

```
\title[short title]{full title}
```

10 AUTHORS AND AFFILIATIONS

Each author must be defined separately for accurate metadata identification. Multiple authors may share one affiliation. Authors’ names should not be abbreviated; use full first names wherever possible. Include authors’ e-mail addresses whenever possible.

Grouping authors’ names or e-mail addresses, or providing an “e-mail alias,” as shown below, is not acceptable:

```
\author{Brooke Aster, David Mehldau}
\email{dave,judy,steve@university.edu}
\email{firstname.lastname@phillips.org}
```

The `authornote` and `authornotemark` commands allow a note to apply to multiple authors — for example, if the first two authors of an article contributed equally to the work.

If your author list is lengthy, you must define a shortened version of the list of authors to be used in the page headers, to prevent overlapping text. The following command should be placed just after the last `\author{}` definition:

```
\renewcommand{\shortauthors}{McCartney, et al.}
```

Omitting this command will force the use of a concatenated list of all of the authors’ names, which may result in overlapping text in the page headers.

The article template’s documentation, available at <https://www.acm.org/publications/proceedings-template>, has a complete explanation of these commands and tips for their effective use.

Note that authors’ addresses are mandatory for journal articles.

11 RIGHTS INFORMATION

Authors of any work published by ACM will need to complete a rights form. Depending on the kind of work, and the rights management choice made by the author, this may be copyright transfer, permission, license, or an OA (open access) agreement.

Regardless of the rights management choice, the author will receive a copy of the completed rights form once it has been submitted. This form contains \LaTeX commands that must be copied into the source document. When the document source is compiled, these commands and their parameters add formatted text to several areas of the final document:

- the “ACM Reference Format” text on the first page.
- the “rights management” text on the first page.
- the conference information in the page header(s).

Rights information is unique to the work; if you are preparing several works for an event, make sure to use the correct set of commands with each of the works.

The ACM Reference Format text is required for all articles over one page in length, and is optional for one-page articles (abstracts).

Table 2. Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

12 CCS CONCEPTS AND USER-DEFINED KEYWORDS

Two elements of the “acmart” document class provide powerful taxonomic tools for you to help readers find your work in an online search.

The ACM Computing Classification System — <https://www.acm.org/publications/class-2012> — is a set of classifiers and concepts that describe the computing discipline. Authors can select entries from this classification system, via <https://dl.acm.org/ccs/ccs.cfm>, and generate the commands to be included in the \LaTeX source.

User-defined keywords are a comma-separated list of words and phrases of the authors’ choosing, providing a more flexible way of describing the research being presented.

CCS concepts and user-defined keywords are required for for all articles over two pages in length, and are optional for one- and two-page articles (or abstracts).

13 SECTIONING COMMANDS

Your work should use standard \LaTeX sectioning commands: `section`, `subsection`, `subsubsection`, and `paragraph`. They should be numbered; do not remove the numbering from the commands.

Simulating a sectioning command by setting the first word or words of a paragraph in boldface or italicized text is **not allowed**.

14 TABLES

The “acmart” document class includes the “booktabs” package — <https://ctan.org/pkg/booktabs> — for preparing high-quality tables.

Table captions are placed *above* the table.

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper “floating” placement of tables, use the environment `table` to enclose the table’s contents and the table caption. The contents of the table itself must go in the `tabular` environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on `tabular` material are found in the *\LaTeX User’s Guide*.

Immediately following this sentence is the point at which Table 2 is included in the input file; compare the placement of the table here with the table in the printed output of this document.

To set a wider table, which takes up the whole width of the page’s live area, use the environment `table*` to enclose the table’s contents and the table caption. As with a single-column table, this wide table will “float” to a location deemed more desirable. Immediately following this sentence is the point at which Table 3 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

Table 3. Some Typical Commands

Command	A Number	Comments
<code>\author</code>	100	Author
<code>\table</code>	300	For tables
<code>\table*</code>	400	For wider tables

Always use `midrule` to separate table header rows from data rows, and use it only for this purpose. This enables assistive technologies to recognise table headers and support their users in navigating tables more easily.

15 MATH EQUATIONS

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

15.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin . . . \end` construction or with the short form `$. . . $`. You can use any of the symbols and structures, from α to ω , available in \LaTeX [?]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n \rightarrow \infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

15.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in \LaTeX ; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n \rightarrow \infty} x = 0 \quad (1)$$

Notice how it is formatted somewhat differently in the **display-math** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \quad (2)$$

just to demonstrate \LaTeX 's able handling of numbering.

16 FIGURES

The “figure” environment should be used for figures. One or more images can be placed within a figure. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.

Fig. 4. 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (<https://goo.gl/VLCRBB>).

Your figures should contain a caption which describes the figure to the reader.

Figure captions are placed *below* the figure.

Every figure should also have a figure description unless it is purely decorative. These descriptions convey what's in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than 2000 characters long (including spaces). **Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper.** For figures that convey important and complex new information, a short text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see <https://www.acm.org/publications/taps/describing-figures/>.

16.1 The “Teaser Figure”

A “teaser figure” is an image, or set of images in one figure, that are placed after all author and affiliation information, and before the body of the article, spanning the page. If you wish to have such a figure in your article, place the command immediately before the `\maketitle` command:

```
\begin{teaserfigure}
  \includegraphics[width=\textwidth]{sampleteaser}
  \caption{figure caption}
  \Description{figure description}
\end{teaserfigure}
```

17 CITATIONS AND BIBLIOGRAPHIES

The use of \LaTeX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names (“Donald E. Knuth”) not initials (“D. E. Knuth”) — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the `\end{document}` command:

```
\bibliographystyle{ACM-Reference-Format}
```

`\bibliography{bibfile}`

where “bibfile” is the name, without the “.bib” suffix, of the \TeX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the “author year” style; for these exceptions, please include this command in the **preamble** (before the command “`\begin{document}`”) of your \LaTeX source:

`\citestyle{acmauthoryear}`

Some examples. A paginated journal article [?], an enumerated journal article [?], a reference to an entire issue [?], a monograph (whole book) [?], a monograph/whole book in a series (see 2a in spec. document) [?], a divisible-book such as an anthology or compilation [?] followed by the same example, however we only output the series if the volume number is given [?] (so Editor00a’s series should NOT be present since it has no vol. no.), a chapter in a divisible book [?], a chapter in a divisible book in a series [?], a multi-volume work as book [?], a couple of articles in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [? ?], a proceedings article with all possible elements [?], an example of an enumerated proceedings article [?], an informally published work [?], a couple of preprints [? ?], a doctoral dissertation [?], a master’s thesis: [?], an online document / world wide web resource [? ? ?], a video game (Case 1) [?] and (Case 2) [?] and [?] and (Case 3) a patent [?], work accepted for publication [?], ‘YYYYb’-test for prolific author [?] and [?]. Other cites might contain ‘duplicate’ DOI and URLs (some SIAM articles) [?]. Boris / Barbara Beeton: multi-volume works as books [?] and [?]. A couple of citations with DOIs: [? ?]. Online citations: [? ? ?]. Artifacts: [?] and [?].

18 ACKNOWLEDGMENTS

Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

This section has a special environment:

`\begin{acks}`

...

`\end{acks}`

so that the information contained therein can be more easily collected during the article metadata extraction phase, and to ensure consistency in the spelling of the section heading.

Authors should not prepare this section as a numbered or unnumbered `\section`; please use the “acks” environment.

19 APPENDICES

If your work needs an appendix, add it before the “`\end{document}`” command at the conclusion of your source document.

Start the appendix with the “appendix” command:

`\appendix`

and note that in the appendix, sections are lettered, not numbered. This document has two appendices, demonstrating the section and subsection identification method.

20 SIGCHI EXTENDED ABSTRACTS

The “sigchi-a” template style (available only in \LaTeX and not in Word) produces a landscape-orientation formatted article, with a wide left margin. Three environments are available for use with the “sigchi-a” template style, and produce formatted output in the margin:

- `sidebar`: Place formatted text in the margin.
- `marginfigure`: Place a figure in the margin.
- `marginfigure`: Place a figure in the margin.
- `marginfigure`: Place a figure in the margin.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

A RESEARCH METHODS

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B ONLINE RESOURCES

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.