The ABCDE Format
Enabling Semantic Conference Proceedings

Anita de Waard[1,2] and Gerard Tel[2]

[1] Advanced Technology Group
Elsevier, The Netherlands
anita@cs.uu.nl
[2] Department of Computer Science
Utrecht University, The Netherlands
gerard@cs.uu.nl

Core Matter

- We believe that the best way to present a narrative to a computer is to let the author explicitly create a rich semantic structure for the article during writing.
- We propose an open-standard, widely (re)useable format, the ABCDE format for proceedings and workshop contributions that can be easily mined, integrated and consumed by semantic browsers and wikis.
- There need not be an abstract in an ABCDE document - instead, the author denotes core sentences within the B,C and D sections, which are compiled through a macro to form a structured abstract.
- We believe a LaTeX stylesheet provides a suitable input format for providing authors with a semantic structure to work from.
- We provide the `abcde.sty` LaTeX file as an appendix to this paper.
- Macros are provided to specify Dublin Core Elements, and to print a list of those that are specified.
- Our section division into Background, Contribution, and Discussion is backed by a number of emperical studies.
- We aim to work on different incarnations of this format and open it up to modification and development.

b1 Introduction

The main problem with automatically extracting information from scientific articles is that the genre of the scientific publication has developed to be an indivisible information unit [1]. The scientific paper is a self-contained narrative, created anew in each iteration, with specific genre characteristics that minimize the potential of identification, content reuse and knowledge integration. All this rhetorical freedom comes at the expense of usability in a computer-centered environment. The linear narrative was fine when we still read and wrote on paper,

but the digital environment in which scientists live and work today calls for a new fundamental unit of communication.

We believe that the best way to present a narrative to a computer is to let the author explicitly create a rich semantic structure for the article during writing. As conceptual structures become the central bearer of information, a set of structured documents can be integrated to form a knowledge network, or structured package of related knowledge regarding a topic [2]. This can be seen to form an incarnation of the intelligent data ideal, which the Semantic Web is meant to enable[3]. The purpose of our work is to examine such a new form of structured publications. Semantic Browsers such as PiggyBank[4] and semantic collaborative authoring tools such as Semantic Wiki's – as presented at this workshop[5] – are paving the infrastructural road for distributed, semantic communities to communicate. Hopefully, the ABCDE format can be a useful vehicle on this road.

Article Outline. This paper is organised as follows: in Section c2, the ABCDE format is described and motivated; in Section c3, the annotation and rendering of ABCDE articles in LaTeX is described. In Section d4, we discuss related work and in Section d5 some next steps.

The section numbers are consecutive, but are prefixed by a modifier: b, c, or d. These are meant as a visual cue to reflect whether the section is a part of the Background, Contribution or Discussion content of the article (described below). The reason for adding this modifier is to help the reader, but also the author, to realise which part is which - if desired, the stylesheet can be modified to make this formatting aspect invisible to humans, and only visible to computers.

c2 The ABCDE Format

We propose an open-standard, widely (re)useable format, the ABCDE format for proceedings and workshop contributions that can be easily mined, integrated and consumed by semantic browsers and wikis. It is characterised by marking the following elements in a document:

Annotations: Each record contains a set of metadata that follows the Dublin Core standard[6]. This metadata is included as a part of each paper, to alleviate the annoying experience that one encounters when an article is found floating in cyberspace, without a date or any bibliographic reference information. In this sense, the DC qualifiers act as a passport that identifies the paper's date and place of birth, for future readers. Minimally required fields are Title, Creator, Identifier and Date. They can be rendered as a part of the text (see below) or left only as mark up, and not printed.

[3] "The Semantic Web is not so much about intelligent agents, but more about stupid agents and intelligent data", Berners-Lee at WWW4, Boston, 1995, personal record.
[4] http://simile.mit.edu/piggy-bank/
[5] http://www.semwiki.org/
[6] http://dublincore.org/documents/dces/

Background, Contribution, Discussion: The material in the main body of text is classified in one of three types:

- *Background*, describing the positioning of the research, ongoing issues and the central research question;
- *Contribution*, describing the work the authors have done: any concrete things created, programmed, or investigated;
- *Discussion*, contains a discussion of the work done, comparison with other work, and implications and next steps.

This classification must be made explicit in the metadata of the article – for details, see Section c3 on markup below.

Entities: Throughout the text, entities such as references, personal names, project websites, etc. are identified inside LaTeX as footnotes or references. The entities can be mined and turned into RDF, where the triple contains the section of the paper containing the entity, the entity URI, and the type of link (reference, person, project).

Identifying the contribution type will increase the quality of the property that can be inferred. For example, the mention of a project website in the Contribution probably means that the project is one of the core components of the system described in the paper. On the other hand, a project website mentioned in the Discussion probably means it is described as a Related Work.

Core sentences as abstract: There need not be an abstract in an ABCDE document - instead, the author denotes core sentences within the B,C and D sections, which are compiled through a macro to form a structured abstract. Upon retrieval or rendering of the article, these can be extracted to form a structured abstract of the article. This allows the author to create and modify statements summarising the article only once, and prevents that an abstract misrepresents the content of the article. This can easily happen when sections are deleted from the content, but left in the abstract, as was shown in [3] A Core summary also enables the implementation of a structured, hyperlinked abstract, where one can jump directly to the relevant part of the article from the sentence of interest.

c3 How to semantically mark your paper

We believe a LaTeX stylesheet provides a suitable input format for providing authors with a semantic structure to work from. The `abcde.sty` style file implements the ABCDE structure for documents typeset with Springer's LaTeX `llncs.cls` class file, very commonly used for proceedings publications in computer science. We provide the `abcde.sty` LaTeX file as an appendix to this paper.

The LaTeX style sheet. To create a consistent layout for its proceedings, Springer makes available to authors and editors its class file `llncs.cls`[7]. It provides

[7] http://www.springer.com/sgw/cda/frontpage/0,11855,5-164-2-72376-0,00.html

the common sectioning commands `\section{..}` through `\paragraph{..}` and some theorem-like environments (`\begin{theorem} ... \end{theorem}`). To authors, the `llncs` class ressembles the common `article` class, but it is richer in marking the contribution with semantic metadata. Specifically, besides `\title` and `\author` commands, there are a `\titlerunning`, `\subtitle`, `\email`, and `\institute` commands.

If your paper was prepared with the `llncs` LaTeX package, to semantically mark it with the ABCDE format:

- Store the style file `abcde.sty` in the same folder as the paper;
- Add the line `\usepackage{abcde}` to your preamble.

Keep in mind that the purpose of semantic marking is mainly to produce meta information that goes with the document. We have chosen to render some of the markup as visual elements as well (for example, by prefixing the section numbers with b, c, or d). This was done for illustration purposes, and the style sheet can be modified to make the proposed semantic marking invisible in the printed result.

The command `\tableofcontents` and the environment `abstract` remain available, but you may choose to have the new command `\listofcore` instead (or in addition). This will produce a list of sentences that you have marked as *core* in the paper.

c3.1 Annotations

Macros are provided to specify Dublin Core Elements, and to print a list of those that are specified. A Dublin Core element is characterized by a *name* and a *value*; it can be specified in the contribution by the command `\dublincore{..}{..}` with the name and value as first and second argument. For example, you can place `\dublincore {publisher} {Creative Commons}` anywhere in your document; the preamble would be the most logical place. (For more about Creative Commons, see http://creativecommons.org/about/licenses/.) On the other hand, `\dublincore{subject}{Dublin Core}` would be logically placed at the place where you discuss Dublin Core elements, so that if you decide to remove some material from your paper, the annotation is removed as well.

The annotations can be just used as metadata without being displayed, but a list can be printed anywhere in the document using the command `\annotations`. For this document, the result would be:

DC Annotations
creator: Anita de Waard , Gerard Tel
title: The ABCDE Format
date: May 4, 2006
subject: ABCDE Format
subject: The `llncs.sty` LaTeX style
publisher: Creative Commons
subject: Dublin Core

Some metadata already available in LaTeX-typeset documents is automatically interpreted as a DC element: specifically, the elements **creator**, **title**, and **date** will be registered as DC elements if they are provided in the preamble with the usual commands.

c3.2 Background, Contribution, Discussion, and Core

The commands `\background`, `\contribution`, and `\discussion` declare that the material following it is the Background, the Contribution, or the Discussion of your document. The semantic marking commands do not replace sectioning commands, so you still need to name the sections in your document.

The simplest documents have these three parts consecutively, so they will have just one `\background` command at the beginning, one `\contribution` command after one or two sections, and one `\discussion` command near the end. The `abcde` package allows to switch between the three types more flexibly. The declaration implied by one of the three commands remains valid until the next `\background`, `\contribution`, or `\discussion` command. If your document contains material that does not fall in one of the three types, precede it with the `\unbcd` command.

Important statements can be marked as *core sentences* using the `\core{..}` command; these sentences can be harvested to give an overview of the content of the paper, with the possibility to jump directly to the relevant part of the paper. Using core sentences can replace an abstract, as they become part of the "Core Matter" list produced by the `\listofcore` command; this command was used on the first page of this paper. Of course, authors have the possibility of writing their abstract instead, and have the core sentences only as metadata pointers to their work.

Markup as core does not change the marked sentence visibly. We found that sometimes, sentences do not read well when taken out of their context; this may happen because of anaphors ("This is a result of ...") or because of a more complicated entanglement with surrounding sentences. If the phrasing of the sentence should differ between the text and the Core Matter list, use the form `\core[Sentence1.]{Sentence2.}` to print `Sentence1.` in the document and have `Sentence2.` in the Core Matter. For example, "She worked in Africa." can be put in the core matter as "Streep worked in Africa." using

`\core[She worked in Africa.]{Streep worked in Africa.}`.

The `\listofcore` command with the optional argument `[1]` will restrict the list to core sentences from the Contribution part of your document (and `[3]` will extend it to also contain core sentences outside of BCD-marked parts).

c3.3 Entities

Both the Dublin Core elements (Annotation) and the (embedded) Elements, such as project websites, references, and personal names, can be extracted to port to an RDF-enabled system.

The notion of entities was described in Section c2; we are looking to expand these to the emerging standard RDF–based formats in the future, in collaboration with Semantic Wiki groups.

c3.4 Shortcomings in the package

It is not possible to have complicated macros in a core sentence. The package was not tested against the various class options of `llncs.cls`.

The purpose of semantic marking is harvesting meta information, but the current package also produces a visible effect of the markings. The should be a possibility to switch the visual effects on and off using a style option `[draft]`; while writing draft versions, the author can keep an eye one the markings he already made, but in the final version the semantic marking would only be harvested, not shown.

d4 Related work

There are many fields of research which offer insights, and important contributions, concerning the structuring and annotation of scientific texts. In an attempt to position the ABCDE proposal within this vast landscape, we distinguish two different dimensions of markup. First the elements marked up, which can be *entities* or *document structure* and second, the time of markup, which can be *during authoring* or *post-publication*. Combined, these describe four categories of work, which are consecutively discussed.

Annotation phase: post-publication; marked up: entities Leaving aside entity extraction techniques[8], an interesting body of work from the Open University revolves around the creation of a "sensemaking environment" which allows readers to manipulate, order and annotate documents. With ClaiMaker [4] and ClaimSpotter [5], they aim to create "a system that explicitly model[s] the rhetorical relations between claims in related papers". Readers can create claim-and-relationship triples according to their ontology of rhetorical relations, to make better sense of the corpus of scientific documents. The triples are identified outside the documents themselves, to improve understanding.

Annotation phase: during authoring; marked up: entities There are several Semantic Wiki and blogging initiatives which propose to provide semantic annotations and allow for distributed access using marked up entities. For instance, Karger and Quan [6] propose to use the innate semantics of messages and blogs to generate semantic markup and utilise it in collaborative (decision-making) systems. Mika and Klein [7] transform BibTex files into RDF, and use it to connect and disseminate bibliographic information of a research group. And Oren

[8] Technically, the huge volumes of work in entity identification and text mining belong in this category, but since the field is vast and not directly related to our research, we will omit a discussion here.

et. al [8] define six dimensions of annotation context, which helps create a faceted browsing interface to improve navigation through their Wiki environment.

In all three intiatives, the author adds (nominal) markup to improve the (RDF-based) metadata of the article.

Annotation phase: post-publication; marked up: text structure Simone Teufel applies a 'rhetorically defined annotation scheme', consisting of seven categories which model prototypical academic argumentation [9]. She first lets a human annotator apply one of seven 'rhetorical roles' to specific elements of a text, and uses this input to train an automatic annotator, which is then used for automated abstract generation.

Noriko Kando [10] defines a fine-grained 'text-level structure', and manually annotates a corpus of (Japanese) articles on HIV/AIDS with this structure. He finds a improved results for searching, passage extraction and browsing tasks.

Bayerl [11] adds three types of markup to a corpus of articles in psychology and linguistics, and then compares the results of the markup. Her three types are structural, which she bases on Kando's schema (above); thematic, based on the Van Dijk Macrostructure concept [12]; and rhetorical, for which she uses Rhetorical Structure Theory [13].

Annotation phase: during authoring; marked up: text structure In contrast, there is much less literature on building systems that allow authors of scientific publications to add markup while creating the text. Quite possibly, this is because this has traditionally been the domain of the publisher, whose methods are proprietary and not under scientific investigation.

Our section division into Background, Contribution, and Discussion is backed by a number of emperical studies.

Kando [10] did an analysis of 40 writing manuals, and came up with a text–level structure where the main headings are Problems, Evidence and Answers. Harmse and Kircz [14] performed a thorough investigation of a corpus of documents in atomic physics, and derived a set of modules, of which the three main ones are Position, Results and Interpretation.

Many journal style guidelines contain a similar division. For example, the American Institute of Physics[9] recommends using Introduction, Main Body and Conclusion as three essential parts of the paper. In the life sciences, papers are often explicitly structured in a similar way: for example, the journal Cell[10] proscribes the sections Introduction, Results, Discussion (and Experimental Procedures) and BMC Cell Biology[11] requires Background, Results, Discussion, Conclusions (and Methods). All of these tripartite divisions correspond quite well to our sections Background, Contribution and Discussion.

The idea of letting authors create markup was motivated in part by the work of Kircz and Harmsze [14], who identified a set of modular elements for a

[9] http://www.aip.org/pubservs/style/4thed
[10] http://www.cell.com/misc/page?page=authors
[11] http://www.biomedcentral.com/bmccellbiol/ifora/

corpus of papers in physics, and created authoring instructions for this modular layout. Work done at Elsevier on the online encyclopedia XPharm[12] built on these explorations, by offering a modular authoring environment, using Word templates.

Structured abstracts Structured abstracts are used in various settings: in medicine, they are quite common (for instance JAMA[13] and the BMJ use them) and are even a topic of study in themselves [15] — since the quality of abstracts is sometimes found to be seriously deficient [3].

Van der Tol [16] has proposed to use structured abstracts as a navigational tool, which could correspond to the 'Core matter' approach, when expanded with the right interface.

d5 Next steps

We aim to work on different incarnations of this format and open it up to modification and development. The point is to offer a flexible structure that can live on semantic environments such as Semantic Wikis and browsers, such as Haystack[14] or Piggybank[15]. Ideally, ABCDE papers should be much easier to mine and integrate. By adding semantic markup of knowledge elements, discovery and integration of information at a structural level is improved.

It is our aim to manually mark up (a subset of) the papers presented to the SemWiki[16] workshop in LaTeX with the `abcde.sty` stylesheet, and open it up for testing before, during and after the workshop. We actively seek collaboration with groups working on Semantic Wikis to see if the format is indeed suitable for transformation to RDF, and how the the metadata can be optimally mined, stored and visualized. The ideal is to narrow the gap between publications and annotations, between doing research and talking about it. We mean to practice what we preach, and will attempt to use the ABCDE format for all relevant conference submissions. Hopefully, with concomitant RDF database and interface work, we can create a 'tipping point'[17] for the implementation of this format, and contribute to the creation of a much richer set of conference proceedings in computer science.

An example of possible developments would include the creation of a conference program, consisting of 'core-contribution' sentences, that link to contributions, as a quick way to scroll around the papers presented. Another example would be to mine all the links to a project website and connect them to the website, linked to the paragraph where the project was mentioned. This would allow the visualization of related projects, topics and co-authors.

[12] http://www.xpharm.com
[13] http://jama.ama-assn.org/
[14] http://haystack.lcs.mit.edu/
[15] http://simile.mit.edu/piggy-bank/
[16] http://semwiki.org
[17] http://www.gladwell.com/tippingpoint/index.html

For example, in the OpenAcademia.org project [7], BibTex references are turned into RDF to allow a connected set of bibliographic references utilising an Open Source extension of RSS called BuRST[18]. This rendering could be used to mine the references of an ABCDE paper, as well — and include the section of the text where the reference was made, again enhancing the quality of inferrable information. Again, the division between a paper and a discussion of a reference begins to blur, and the publication itself can become a (set of) object(s) on a Semantic Web/Wiki.

Our further work will involve the development of a more detailed model of scientific publications, and looking at the construction of meaning within scientific documents through argumentation analysis and understanding of discourse structure. The tension between the arguments or moves and the narrative of the document as a whole poses an interesting topic of study in terms of both knowledge modeling and rhetoric/discourse studies. Hopefully, it can also help create a more legible way to publish science for computer-assisted humans, and human-assisted computers.

6 Acknowledgement

We wish to thank Simon Pepping of Elsevier for helping think through the ABCDE format.

References

1. Bazerman, C.: Shaping written knowledge: The genre and activity of the experimental article in science. Madison: University of Wisconsin Press (1988)
2. de Waard, A.: Science publishing and the semantic web, or: Why are you reading this on paper? European Conference on the Semantic Web 2005, Industry Forum, Alain Léger ed. (2005)
3. Pitkin, R.M., Branagan, M.A., Burmeister, L.F.: Accuracy of data in abstracts of published research articles. JAMA **281** (1999) 1110–1111
4. Li, G., Uren, V., Motta, E., Buckingham-Shum, S., Domingue, J.: Claimaker: Weaving a semantic web of research papers (2002)
5. Shum, S.B., Domingue, J., Motta, E.: Scholarly discourse as computable structure. In: OHS-6/SC-2. (2000) 120–128
6. Karger, D.R., Quan, D.: What would it mean to blog on the semantic web? Journal of Web Semantics **3**(2) (2005)
7. P. Mika, M. Klein, R.S.: Semantics-based publication management using RSS and FOAF. In: Proceedings, Semantic Wiki 2006 (Submitted). (2006)
8. Oren, E., Delbru, R., Möller, K., Völkel, M., Handschuh, S.: Annotation and navigation in semantic wikis. In: SemWiki (ESWC). (2006) Submitted.
9. Teufel, S., Moens, M.: Discourse-level argumentation in scientific articles: Human and automatic annotation. In Walker, M., ed.: Towards Standards and Tools for Discourse Tagging: Proceedings of the Workshop. Association for Computational Linguistics, Somerset, New Jersey (1999) 84–93

[18] http://www.cs.vu.nl/ pmika/research/burst/BuRST.html

10. Kando, N.: Text-level structure of research papers: Implications for text-based imformation processing systems (1997)
11. Bayerl, P.S.: Methods for the semantic analysis of document markup (2003)
12. Dijk, T.: Macrostructures: An interdisciplinary study of global structures in discourse, interaction, and cognition. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1980)
13. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text **8**(3) (1998) 243–281
14. Kircz, J., Harmsze, F.: Modular scenarios in the electronic age. In: CS-Report 00-20. Proceedings Conferentie Informatiewetenschap 2000. De Doelen Utrecht, 5 april 2000. (2000)
15. Wong, H., Truong, D., Mahamed, A., Davidian, C., Rana, Z., Einarson, T.: Quality of structured abstracts of original research articles in the british medical journal, the canadian medical association journal and the journal of the american medical association: a 10-year follow-up study. Curr Med Res Opin. **21**(4) (2005) 467–73
16. van der Tol, M.: Abstracts as orientation tools in a modular environment. Document Design **2**(1) (2001) 76–88

A The `abcde.sty` style sheet

```
%
% File: abcde.sty
% Created 16/03/2006 by Gerard Tel
% Defines semantic annotations following the ABCDE proposal
% for documents already in llncs format.
%
%
% A is for Annotations
% An annotation in the Dublin Core format has two characteristics:
% the NAME of the element and its VALUE.
% A possible third argument is the SCHEME describing the VALUE format.
% Make a Dublin Core element using \dublincore[scheme]{name}{value}
% to write name and value to file dce:
\newcommand\dublincore[3][DEFAULT SCHEME]{
\addcontentsline{dce}{dcelt}{{#2}{#3}}}
%
% The DC annotations can, but need not, be printed in the text:
\def\annotationsname{DC Annotations}
% How to print a name/value combination:
\def\core@nameval#1#2{{\bf #1:} #2}
\def\@dceltline#1#2{% #1 = {name}{value}, #2 = pageno
  \par\noindent\core@nameval#1 \par}
\def\l@dcelt{\@dceltline}
% Print the core sentences with \annotations
\newcommand\annotations{%
  {\noindent\bf\annotationsname}\par
  \@starttoc{dce}}
```

```
%
% Harvesting the annotations
% Grab as much as you can from \maketitle:
\let\orig@maketitle=\maketitle
\renewcommand\maketitle{
  % within the DC element, \and and \inst#1 have different meaning
  \let\orig@and=\and\def\and{, }
  \let\orig@inst=\inst\def\inst##1{}
  \dublincore{creator}{\@author}
  \def\and{\orig@and}\def\inst{\orig@inst}
  \dublincore{title}{\@title}
  \dublincore{date}{\@date}
\orig@maketitle}

% There are four BCD-types.  Initially it is bcdu (undefined)
\newcommand{\bcd@type}{bcdu}
% The \background, \contribution, and \discussion commands will
% 1. Change the section numbering by prefixing a letter
% 2. Change the abs entries by changing \bcd@type
\newcommand{\bcd@swap}[2]{
  \def\thesection{#1\arabic{section}}
  \def\thefigure{#1\arabic{figure}}
  \def\bcd@type{#2}}
\newcommand{\background}{\bcd@swap{b}{back}}
\newcommand{\contribution}{\bcd@swap{c}{cont}}
\newcommand{\discussion}{\bcd@swap{d}{disc}}
\newcommand{\unbcd}{\bcd@swap{}{bcdu}}
%
%
% The command \core{TEXT} will print TEXT and save it in the file
% basename.abs as a contentsline
% Two args, first is optional with default second
% Print argument normally in text:
\newcommand{\core}[2][\undefined]{
\ifx\undefined#1#2\else#1\fi
\addcontentsline{abs}{\bcd@type}{#2}}
\def\listofcorename{Core Matter}
\def\@coreline#1#2#3{% #1: importance, #2: text, #3: Page no
  \ifnum#1>\c@coredepth
  \else \item#2\vskip 3\p@\fi}
\def\l@bcdu{\@coreline{3}} % Relatively unimportant
\def\l@back{\@coreline{2}} % Moderately important
\def\l@cont{\@coreline{1}} % Important
\def\l@disc{\@coreline{2}} % Moderately important
% Print the core sentences with \listofcore[i],
```

```
% Where i=1 prints only contribution core, 2 prints B&D as
% well, and i=3 prints "unlabeled" core.
\newcommand\listofcore[1][2]{%
  \def\c@coredepth{#1}
  \@restonecolfalse\if@twocolumn\@restonecoltrue\onecolumn\fi
  \section*{\listofcorename\@mkboth{{\listofcorename}}{{\listofcorename}}}
  \begin{itemize}\@starttoc{abs}\end{itemize}
  \if@restonecol\twocolumn\fi}
```