Improved Learning of Academic Writing
Reducing Complexity by Modeling Academic Texts

Gert Faustmann
Department of Cooperative Studies, Hochschule für Wirtschaft und Recht Berlin (Berlin School for Economics and Law), Alt-Friedrichsfelde 60, 10315 Berlin, Germany

Keywords: Text Coherence, Writing Process, Argument Representation, Text Perspectives, Text Patterns.

Abstract: A graphical modeling language for scientific texts is presented, which particularly supports the learning process of academic writing. The supervision process for textual work in higher education is often characterized by misunderstandings, since the agreements are based on the abstract level of the document outline. The written text then often misses an inner structure and suitable representations and thus has little coherence. Since an academic text is a complex combination of text segments, linguistic functions, content, and means of presentation, a notation is proposed that includes these different perspectives. Based on the UML for software modeling, extracts of a text as well as different levels of abstraction can so be part of the learning process. Finally, a software tool is sketched, which can support the construction of document frameworks as well as the creation of the actual text.

1 INTRODUCTION

In scientific discourse, publications are a fundamental part of communication. The quality of scientific texts is not only determined by the quality of the research work carried out, but also by the clarity of the presentation of the results obtained. This property of texts is called coherence (Beaugrande and Dressler, 1981) and expresses itself in scientific texts as an inner connection of argumentations and the parts contained therein (statement, evidence, justification, qualification) (Booth et al., 2003).

Academic papers at universities are examinations in all phases of the study programme. As the course progresses, the complexity of the texts to be produced increases. The creation process is not only influenced by the student, but also by an academic supervisor during the planning process. However, misunderstandings often arise here, as the subject of the discussion is more likely to be at the abstract level of the document structure (which means the outline with chapters and subsections). Concrete content-related characteristics of the individual sections in the text only become clear after submission of the work and often exhibit a lack of coherence.

There is therefore a gap between the planning of a scientific text at its structure level (document structure) and its content characteristics (functional structure). A well-arranged representation of this functional structure in between could on the one hand make the support process in teaching more efficient and on the other hand fundamentally improve text production in the academic field.

2 SUPPORT FOR ACADEMIC WRITING PROCESSES

The necessity of supporting writing processes is recognised in university education and is also implemented by various measures. Considering the current situation at universities, there are common approaches to support academic writing. However, various authors also suggest more extensive means to improve the process of writing and supervising a thesis. One of these is stronger planning and modeling.

2.1 Common Approaches

In order to promote the writing process at universities and colleges, three approaches can be distinguished:

- In *courses on scientific work and academic writing*, behaviors are taught to plan a written work, to collect information and to find a question whose solution with suitable representations results in a text. In this context, textbooks dealing with academic writing are also used.

Faustmann, G.
Improved Learning of Academic Writing.
DOI: 10.5220/0006792204470453
In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018)*, pages 447-453
ISBN: 978-989-758-291-2

- Through *examples of dissertations* (e. g. from the university library) students can see how concrete topics were worked on and specific problems were solved in detail (e.g (Coffin et al., 2002) describing an argument essay outline).
- In specific *support processes of written dissertations* research questions are chosen and formulated, approaches and structures are discussed, as well as formal criteria are clarified. Text excerpts, graphic representations, structures and abstracts are used for communication.

If there is a writing task in the course of studies, students are confronted with various *problems*:

- On the one hand, the contents of the courses on scientific writing date back a long time and are no longer directly available as practical knowledge. In addition, most of the contents were not focused enough on the current question of the work to be done.
- The sample papers provide an insight into the design and framework conditions of a final thesis, but they usually cannot be mapped to the question, method or argumentation of the student's own task.
- Due to the formal framework conditions, such as limited time for discussion and only rough descriptions of the text to be written, the support process remains at a rather abstract level.

2.2 External Document Representations

In general, the use of external representations of thought processes on different levels is recognized as the basis for a qualitatively better design performance (Kirsh, 2010).

Various forms of externalisation have found their way into scientific text production in recent years. Mindmaps are generally known (Buzan, 2006) and serve the hierarchical organization of questions. They can also be easily combined with brainstorming processes. In this context, concept maps are also known, which do not have to be hierarchically structured and combine ideas (concepts) in a network structure (e. g. (Heard, 2016)).

The general explication of argumentation was already described in early studies (Newell, 1979) (Dijk, 1980) (Toulmin, 2003). Some authors design specific models for argumentation chains in e-learning ((Bell, 1997) (Faustmann, 2011) and in hypertext systems (Neuwirth and Kaufer, 1989) (Streitz et al., 1992).

Hausdorf examines, for example, the entire development process of a scientific work with information objects involved in it (Hausdorf, 2005). The text structure is also discussed here and in the implemented software tool ScientiFix hierarchical representations of sections, ideas and sources can be found (e. g. Figure 7.12, p. 193). In (Byrne and Tangney, 2010) a tool is developed that includes different representations of a document (map view, tree view and text view) based on the framework of Shibata and Hori (Shibata and Hori, 2002) for organizing document parts.

The perspectives on texts used in Shibata and Hori are strongly reminiscent of outlines (hierarchical perspective) and concept maps (map perspective). Both are nowadays still well known means to support the writing of texts. The overall problem of external representations of texts is, that there is no integrated model for planning the document and text structures on all levels that make up an academic text.

3 CONCEPT OF A TEXT MODELING LANGUAGE

The various problems, which are frequently encountered in more extensive written work, show that students have to cope with the complexity of the functional parts of the text and their interrelationships. For this reason, we first want to analyse briefly how the complexity of other systems to be constructed can be mastered.

3.1 Modeling Complex Products

The problem of complexity can be similarly encountered in the construction of software artifacts: a complex software system can no longer be overlooked in its many tasks and arising dependencies. In recent decades, various paradigms have been established to structure a system (e. g. object orientation) and to plan clearly with suitable representation methods (e. g. Unified Modeling Language UML).

In the design, two types of presentation can be distinguished in principle: on the one hand, the logical design, which describes the task of the system and the subject-specific solution to the problem, and on the other hand, the system design, which is to represent an exact model of the later system. The advantage here lies in the separation of the algorithmic logic from technical decisions: both interact with each other, but should initially be unaffected by each other. The UML not only offers the possibility to display both design layers, but also to convert them into each other without any problems. For example, the Domain Driven Design methodology implements this approach in a software process model (Evans, 2003).

In addition, different views of the architecture of a software system can be described in UML: typical examples are the structural and behavioral view. In this way, class diagrams can describe the basic structure of a system and component diagrams can depict the division of classes into modules. Other diagrams exist, for example, for offered functions (use case diagrams), the physical distribution to host nodes (deployment diagrams) and communication between the objects of the system (sequence diagrams) (see Figure 1).

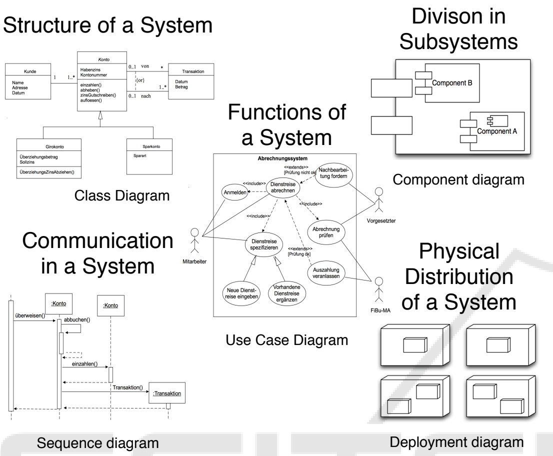

Figure 1: Perspectives on Software in UML.

These different levels of abstraction thus lend themselves to being transferred to text production.

3.2 Requirements for Academic Text Modeling

3.2.1 Ease of Use

To reduce the complexity of a product, it is necessary to reduce the amount of notational elements of a system to a minimum. This does not mean that the models will only have a small scope, but the learning effort for using the language elements will be considerably reduced. Thus the first large requirement area of a text modeling language is *simple use*. This includes being able to grasp a first model relatively quickly and, if necessary, to sketch a model by hand.

3.2.2 Different Perspectives

The next requirement concerns *visibility of different perspectives* on the text to be made. Well-known perspectives are the text structure with its outline, the representations within texts by continuous text, figures, tables, etc., textually described parts of the later text, as well as functions of the text components such as analyses, hypotheses, summaries, etc.

3.2.3 Relationships

Furthermore, it must be possible to create *relationships* between text elements, both within and across perspectives. Examples of relationships are the local sequence in the text (e.g. chapter 2 follows chapter 1), the referencing of parts of the text that are contained earlier in the text and will only appear later (e.g. the comparison of an examination of a foreign author with own results) and the inclusion of a part of the text by a function (e.g. the above-mentioned comparison can be an evidence for a certain hypothesis).

3.3 Language Description

The proposed language for modeling academic texts contains various notation elements, each of which is assigned to one of four perspectives. An important perspective is that of the functional parts, which must meet the requirements of scientific proof. Often, works also show shortcomings in the presentation of the results, to which an own perspective is responding. Finally, relationships between the different text elements indicate dependencies at all levels. See the now detailed language elements in an overview model in Figure 2.

3.3.1 Perspectives

The selection of the contained perspectives is based on two basic conditions: on the one hand, in today's writing processes different means are already used to structure and design texts (see Section 2.1). These include structures, collections of ideas and concept maps. On the other hand, the main problems lie in the preparation of scientific papers in elaboration of the argumentation, as well as in the appropriate presentation of the results (see introduction). This leads to the following four perspectives:

1. Outline Perspective
   This section describes the division of the text into chapters, subchapters and sections. It becomes clear which parts follow each other or contain other parts of the text.
2. Functional Perspective
   Here, functional units are identified and their interrelationships are recorded by appropriate relationships. From this perspective, it becomes clear which descriptions and examples have led to a hypothesis, for example, and which analyses support a hypothesis.

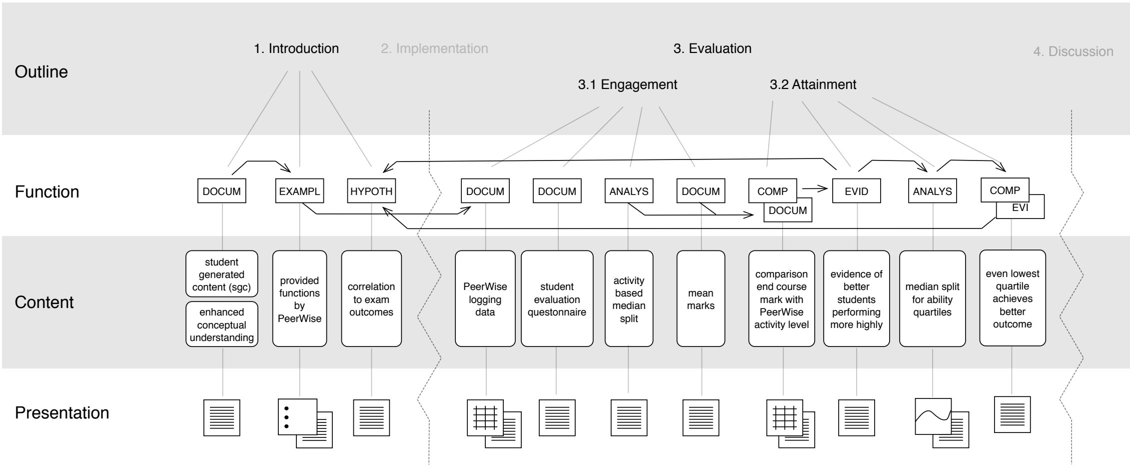

Figure 2: Example for a Text Model.

3. Content Perspective
   In this perspective, short descriptions of the content are assigned to the text elements, which can give the viewer a first idea of the content. These can be compared with concepts in concept or mind maps.
4. Presentation Perspective
   Text elements can be presented very differently. Thus, an analysis of different methods can be done in tabular form or in continuous text. Further known types of presentation are figures, mathematical representations (e.g. equations) and program listings or algorithms.

3.3.2 Functional Elements

For scientific work, functional elements of a text can be subdivided into the areas of knowledge collection/preparation, knowledge creation and knowledge classification (Hausdorf, 2005).

1. The knowledge collection includes

   Collection/Documentation. A selection of the objects to be viewed (DOCUM) or an example representation (EXAMPL) is made.

   Analysis. How is an object broken down into its components (ANALYS)?

   Comparison. Which parts of an object are considered important (COMP)?

2. Knowledge creation is characterized especially by reasoning which contains the following components:

   Hypothesis. What can be assumed on the basis of previous considerations (HYPOTH)?

   Evidence. What speaks for a hypothesis (EVID)?

   Evaluation. How to assess this argumentation (EVAL)?

3. At knowledge classification we need:

   Summary. What is the path of a hypothesis and what are the consequences of the now proven hypothesis (SUMM)?

   Comparison. What relation does new knowledge have to the already existing knowledge (COMP)?

The function elements listed here appear in the function view as symbols with their respective abbreviations.

3.3.3 Presentational Elements

Scientific texts contain the following types of content presentation:

- Text: a text without additional structuring measures. Typically, a unit of text is grouped together as a paragraph with one thought in it.
- Enumerations or Lists: if different parts of a text of one type are described, they can be separated from each other by dots (point / minus signs) and presented in a structured way for the reader. Enumerations number the text parts.
- Tables: they are mostly used to display figures, but can also contain symbolic information (e.g. identifiers).
- Figures: are of very different types and serve to illustrate connections.
- Program Code/ Algorithms: if procedures or even concrete implementations are to be shown, this can be done through suitable languages of an algorithmic nature.

- Mathematical Expressions: describe numerical relationships.

Presentation elements are indicated in the text model by a pictorial representation. Figure 3 shows the assignment of function elements to screen representations.

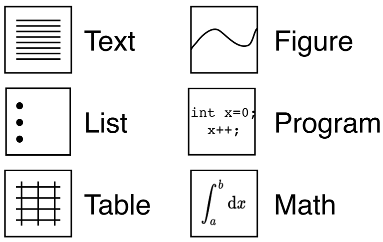

Figure 3: Presentational Elements.

These presentation elements are partly interchangeable. For example, the content of an illustration can also be described in continuous text. On the other hand, however, it makes sense in some places to depict a content to be represented by more than one presentation element (e.g. the content of an illustration should be explained additionally in the text).

3.3.4 Relationships

Relationships between text elements are of different nature: the simplest connection is the *sequence* within the text to be written. Thus chapters follow each other, but also functional units such as an example and an analysis. If the sequence is not clear from the arrangement in the model, a connecting line with an unfilled arrowhead can be used.

Another relation is the *provision* or the *inclusion*. Thus, chapters may include subchapters, but a comparison can also provide evidence for a hypothesis. The relationship is visualized by a simple connecting line.

After all, parts of the text must also be able to relate to each other in their statement. This can be interpreted as a *reference* that may not have any overlapping content. In this way, a later derived evidence can also be used for a hypothesis presented in the text. The reference relationship must be made clear here. This relationship can be shown by a line with an open arrowhead (see Figure 4).

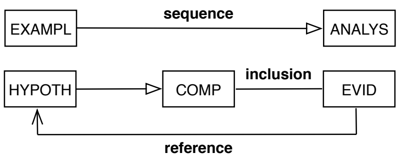

Figure 4: Relationships.

4 CONCLUSIONS

4.1 Advantages of Text Modeling

The approach described here to modeling academic texts explains the argumentative background of a work, as well as the means of presenting the necessary content. This makes it possible to carry out a more detailed planning of a text in terms of its significance before the actual writing process begins.

Furthermore, it is now also clear for existing texts how they are structured without having to be read in detail. It would even be conceivable to offer the essence of a text, i.e. hypotheses of interest to the reader, as well as their proofs by means of documented evidence, in a compact form and to refer only to these passages of the text.

Both possibilities could be used for future learning processes: in the context of supervising dissertations, the construction process of the text could be improved and in the context of lectures, the analysis of existing research reports on the basis of a text model would be much easier to understand for students.

4.2 Limitations

However, the concept also presents difficulties: first of all, the language definition must be available to and understood by all those involved. This involves a certain amount of extra effort, which could be put into perspective within the respective courses and by means of suitable manuals.

One danger in the use of system notations is the supposed security of getting to high-quality systems. The use of engineering methods and notations such as the Unified Modeling Language is by no means a guarantee for the specification of appropriate and error-free software. However, these methods increase the probability of software quality by improving the overview.

Text models can be used to create a framework for scientific texts that contains important parts, but the texts can still be difficult to understand, do not correspond to the assigned functional units and much more.

5 FUTURE WORK

In order to ensure that the concept of modeling academic texts also brings the desired **benefits in practice**, the following additional work is necessary:

- The modeling language must be usable by implementing a first symbol library (e.g. in Microsoft Visio). Of course, handwritten sketches can still be created beyond this.
- The language must be made known so that an application is possible. This includes the presentation in lectures as well as the presentation in e.g. a handbook.
- Typical configurations of functional elements and possibly also presentation elements should be described, which can be used as models for academic text work in teaching. In this case, the modeling language and the formulated patterns also serve as learning materials for general requirements of academic texts. (Figure 5).

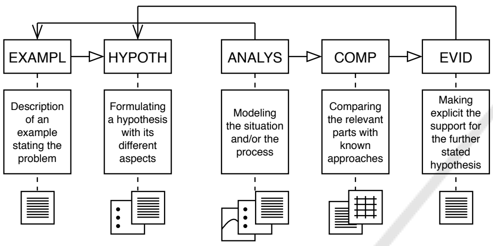

Figure 5: Argumentation Pattern as a Text Model.

- Processes are to be defined which integrate the development of the different design perspectives. Here you will also find references in the literature describing, for example, the integration of outline and content sketches in text planning. The use of patterns or the connection with support processes when creating final theses must also be embedded here.

A further challenge can be seen in the **need for an increased technical support** for students or academics. If it were possible not only to create text models in the four perspectives defined in this paper, but also to integrate the text itself, the creation of a text could be seamlessly designed from the draft to the writing process. Here one could write in different parts of the work, adapt the design and thus achieve an integrative editing with an always good overview for the author.

Existing text creation tools allow to create a first outline of the text and then add the text. Thus, Microsoft Word also offers a so-called Outline View. Other tools go one step further and not only integrate the text with the separate outline representation, but also integrate additional information for the writing process into the document. For example, the tool Scrivener (www.literatureandlatte.com/scrivener) contains a research directory for collecting this information. Scrivener especially offers the management of meta-information in a binder, which refers to the text type Novel (with characterization of figures etc.).

Figure 6 shows an initial draft of a scrivener-like user interface that can integrate the perspectives of a text model. While the well-known outline on the left-hand side of the screen arranges the text's content components hierarchically, the functional and presentation view can be found in the middle columns. Only the section selected in the outline is displayed. From the presentation elements, you can find a direct assignment to the actual text, which is located in the rightmost column. The individual columns can also be hidden by the user.

Finally, there are **other use cases** in the utilization of text models and corresponding software tools conceivable:

- In scientific practice, contributions are often written in groups. This concerns the distribution and coordination of the respective text contributions.

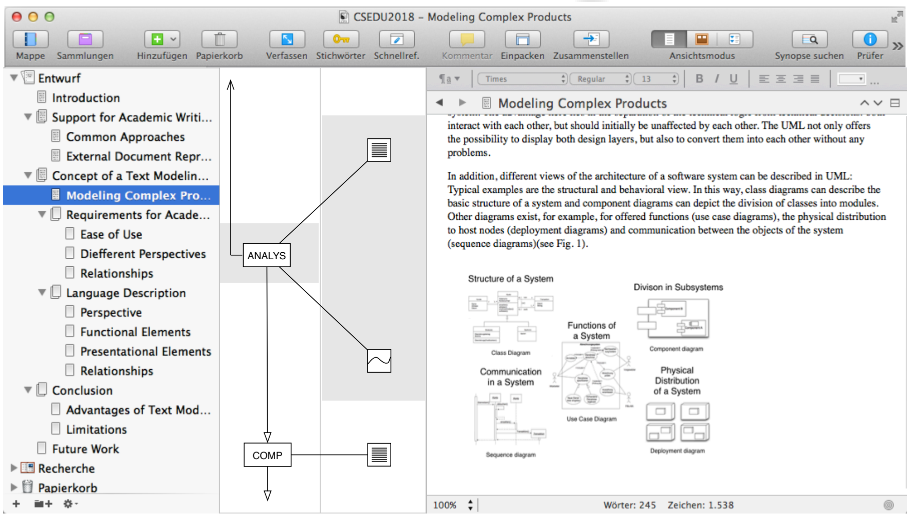

Figure 6: GUI prototype of a writing tool integrating text structure.

It is also conceivable to use formal notation elements that define the expectations on both sides of the respective text parts, similar to interface classes in software systems.

- The processes for the creation of texts with text models have already been addressed and their relation to the support processes. However, it would also be conceivable to have guided learning processes that can be provided by the lecturer with different specifications at all levels (students could not only formulate texts but also analyse them with regard to the functions they contain).
- Text models could not only support the creation of texts, but also the reengineering of existing texts. The aim of such an analysis can be the improvement of texts in their creation process, the understanding of extensive and complex texts and thus also the learning of writing techniques using concrete examples. A software-technical support of the analysis process is conceivable similar to systems for qualitative data analysis (e.g. Atlas/ti, www.atlasti.com). The possibility of greater automation when analysing texts should be investigated.

REFERENCES

Beaugrande, R. B. and Dressler, W. (1981). *Introduction to Text Linguistics*. Routledge, London ; New York.

Bell, P. (1997). Using Argument Representations to Make Thinking Visible for Individuals and Groups. In *Proceedings of the 2nd International Conference on Computer Support for Collaborative Learning*, CSCL '97, pages 10–19, Toronto, Ontario, Canada. International Society of the Learning Sciences.

Booth, W. C., Colomb, G. G., and Williams, J. M. (2003). *The craft of research*. Univ. of Chicago Press, Chicago, Ill. [u.a.].

Buzan, T. (2006). *The Ultimate Book of Mind Maps*. Harper Collins Publishers, London.

Byrne, J. R. and Tangney, B. (2010). CAWriter: a CSCW/CSCL tool to support research students' academic writing. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, pages 458–462, Swinton, UK, UK. British Computer Society.

Coffin, C., Curry, M. J., Goodman, S., and Hewings, A. (2002). *Teaching Academic Writing: A Toolkit for Higher Education*. Routledge, London.

Dijk, T. A. v. (1980). *Macrostructures: an interdisciplinary study of global structures in discourse, interaction, and cognition*. L. Erlbaum Associates, Hillsdale, N.J.

Evans, E. J. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley, Boston.

Faustmann, G. (2011). Project-based Collaborative Learning with Arguments - Flexible Integration of Digital Resources. volume 2, pages 359–364, Noordwijkerhout, The Netherlands. SciTePress.

Hausdorf, C. (2005). *Integratives, ganzheitliches Modell und Werkzeug zur wissenschaftlichen Textproduktion*. PhD thesis, Dissertation. Erlangen-Nürnberg http://www. opus. ub. uni-erlangen. de/opus/volltexte/2005/125/.

Heard, S. B. (2016). *The Scientist's Guide to Writing: How to Write More Easily and Effectively throughout Your Scientific Career*. Princeton Univers. Press, Princeton Oxford.

Kirsh, D. (2010). Thinking with external representations. *AI & SOCIETY*, 25(4):441–454.

Neuwirth, C. M. and Kaufer, D. S. (1989). The Role of External Representation in the Writing Process: Implications for the Design of Hypertext-based Writing Tools. In *Proceedings of the Second Annual ACM Conference on Hypertext*, HYPERTEXT '89, pages 319–341, New York, NY, USA. ACM.

Newell, A. (1979). Reasoning, problem solving and decision processes: The problem space as a fundamental category.

Shibata, H. and Hori, K. (2002). A framework to support writing as design using multiple representations. In *Proc. 5th Asia Pacific Conference on Computer Human Interaction*. Citeseer.

Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schtt, H., and Thring, M. (1992). SEPIA: A Cooperative Hypermedia Authoring Environment. In *Proceedings of the ACM Conference on Hypertext*, ECHT '92, pages 11–22, New York, NY, USA. ACM.

Toulmin, S. E. (2003). *The Uses of Argument*. Cambridge University Press, Cambridge, U.K. ; New York, updated edition.