



## A pattern language for scientific texts – support for the academic education process

Faustmann, Gert; Kirchner, Kathrin; Lemke, Claudia; Monett, Dagmar

*Published in:*

Proceeding of the 14th International Technology, Education and Development Conference

*Link to article, DOI:*

[10.21125/inted.2020.1528](https://doi.org/10.21125/inted.2020.1528)

*Publication date:*

2020

*Document Version*

Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*

Faustmann, G., Kirchner, K., Lemke, C., & Monett, D. (2020). A pattern language for scientific texts – support for the academic education process. In *Proceeding of the 14th International Technology, Education and Development Conference* (pp. 5655-5664). IATED. <https://doi.org/10.21125/inted.2020.1528>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# A PATTERN LANGUAGE FOR SCIENTIFIC TEXTS – SUPPORT FOR THE ACADEMIC EDUCATION PROCESS

Gert Faustmann<sup>1</sup>, Kathrin Kirchner<sup>2</sup>, Claudia Lemke<sup>1</sup>, Dagmar Monett<sup>1</sup>

<sup>1</sup>*Berlin School of Economics and Law (GERMANY)*

<sup>2</sup>*Technical University of Denmark (DENMARK)*

## Abstract

Scientific writing in higher education is not only a prerequisite for participation in scientific discourse, but also forms the basis for the development of one's own thinking. Despite the sometimes extensive writing tasks in various study programs, the results are often unsatisfactory. The reasons for this could be many, but one of the most pressing ones is the difficulties learners face when grasping the complex structure of a scientific paper. Contributing to this might be not only the many components a scientific paper is composed of, but also their often intricately relationships and dependencies.

The central goal of the work presented in this paper is to develop a set of proven structural parts of scientific texts that can be assembled into a model language. Patterns are already used in different contexts to represent typical solutions. Examples can be found in software development and in the optimization of processes. A pattern language can then support the writing process in its different phases by linking the patterns (e.g. from the first creation of the main parts of a report to the detailed argumentation). We suggest mechanisms aimed at both supporting and complementing the creation of a scientific paper. In this regard, we focus on the importance of an early, concrete planning of how selected parts of the desired product, i.e. the intended scientific paper, should be considered.

Three major steps are necessary to achieve that goal: First, the required language elements for the description of texts need to be identified. Typical text patterns can then be formulated with the help of these elements. In order to support the entire writing process, these patterns must finally be linked in a suitable way that allows for the planning and creation of the text in ever increasing degrees of detail. The initial data for determining the language elements as well as the text patterns are provided by the authors' previous experience in the supervision of approx. 2000 final theses, as well as by a thorough analysis of best practices from the literature on scientific writing.

This work marks a starting point in the development of a model language, which is open for the integration of further experiences in the form of new patterns. It also raises the question of how text patterns can be used appropriately in an initial learning process of scientific writing and working.

Keywords: Academic writing, text patterns, pattern language, writing support process.

## 1 INTRODUCTION

Both the large number of textbooks for academic writing available on the market and our own experience in academic settings show that problems often arise for students when writing academic texts during their studies. On the one hand, the motivation to write a long text is a challenge, on the other hand, there is a lack of basic knowledge about the requirements that go beyond simple formal rules. Several courses on scientific writing in tertiary education aim to support students in their work, but in the concrete writing process most formal requirements for doing so are either too general or no longer available in practical situations.

Scientific papers are usually extensive texts, which have to meet many formal and content-related requirements. In our working environment, the volume ranges from ten to fifty pages. The final thesis, which indicates the end of the studies at a higher education level, for example, almost always contains the largest number of pages. For the different types of requirements for a scientific work, there are also various guidebooks: in the formal area for the linguistic design [1], for the research question and the logical structure [2][3], for the type of presentation [4], and even presentations that summarize all these topics [5]. A large area is also devoted to advice on how to overcome motivation difficulties when writing academic texts [6][7].

In the creation of complex systems in areas other than scientific writing, the documentation of best practices has proven to be useful and is usually presented in a way that is suitable for general use.

When formal models are used for representation, as well as a semi-structured notation for the rough structure of the documentation, these representations are called patterns. A pattern is described by its name, the problem addressed, the solution idea, and the consequences of using the pattern [8]. Thus, a pattern can also be used by users who are not yet familiar with best practices but who have a corresponding problem to solve. The possible dangers and disadvantages of using patterns may become clear through the pattern description. Thus, many proven approaches are available in a compact and clear presentation, which can be searched for depending on the basis of the existing problem situation.

The aim of the present work is to elaborate on a first set of patterns for scientific texts. A prerequisite for this is the existence of a description language for the different parts of a scientific text. In addition to the patterns, however, the question also arises as to how they can be integrated into the creation or support process in the context of higher education. In so-called pattern languages, there are references between the patterns, which realize the structure of the object to be created (e.g. the scientific work) by the continuous application of more concrete detailed practices. The explanations will then lead to a corresponding text pattern language, which will support the writing process from the first rough structure to the details of the text.

## 2 FUNDAMENTALS OF PATTERNS IN LEARNING PROCESSES

For the establishment of solution patterns, it is necessary to clarify which components a pattern should have and what function these components have in each case. The use of patterns in learning processes can be divided into two aspects: On the one hand, patterns describe the implementation of the learning itself (which are called “pedagogical patterns”). On the other hand, they refer especially to software artifacts, where there has been a rapid development of so-called design patterns since the late 70s. As concrete support for the writing process, pattern-based approaches can be found in the literature, which lead to the implementation of supporting software systems.

### 2.1 Basic Idea of Patterns

The solution of problems in the engineering sciences is not achieved by constantly reinventing approaches to solutions. In all disciplines there are recognized procedures and approaches that have proven themselves. Examples of such procedures are:

- The design of interiors in **architecture**: The need for seating and proximity to light requires the creation of seats near windows, e.g. in a bay window ([9], as referred in [10]).
- The reduction of complexity by hiding implementation details of a software component in **software engineering**: A facade forwards the calls of detailed services of a component to the responsible subcomponents [8].
- The stable connection of woods in **furniture design**: Depending on the desired connection thickness and various other parameters, e.g. the tenon joint is a solution [11].

For the description of patterns, however, it is not sufficient to describe only the structure of the desired object (for the above examples, the floor plan of the room, the interaction of the classes in a facade or the appearance of tenon joints). It is also necessary to know e.g. what risks are involved in the application and what an example of implementation might look like.

Since design patterns in software engineering have been extensively studied and driven by intensive research, we take a closer look at the parts of a pattern description proposed in this area. In [8] the components of a pattern description for software products are designed in detail. This description contains 14 elements (see Table 1).

Table 1: Attributes that describe software design patterns ([8])

Attribute	Meaning
Name	Uniquely describes a pattern
Classification	Defines a certain class of problems the pattern is assigned to
Purpose	Defines the concrete problems the pattern can be used for
Also known as	Shows other designations of the pattern
Applicability	Shows different situations in which the pattern can be used
Motivation	Scenario in which the pattern can be used
Structure	Shows the single parts of the pattern
Participants	Describes who participates in the pattern's situation
Interactions	Describes the interactions between the participants
Consequences	Shows the advantages and disadvantages when using the pattern
Implementation	Gives hints on how to implement the pattern in source code
Example	Shows some sample code implementing the pattern
Known uses	Shows examples of the implementation in existing software systems
Related patterns	Describes other patterns that are dependent or that can be used alternatively

In addition to the set of patterns, further information is required for their use. In the basic textbooks for design patterns the patterns are described as a coherent system. Buschmann et al. [10] describe a pattern system based on the set of individual patterns (this system Alexander calls a pattern language in [12]):

"A pattern system for software architecture is a collection of patterns for software architecture, together with guidelines for their implementation, combination and practical use in software development."

In order to make use of patterns, it is therefore necessary to provide, in addition to the actual patterns, information about their combination and use in the respective phase of a development process. Phase-oriented categories that can be assigned to the respective development stages are useful here. Buschmann et al. [10] define, for example, the categories architecture patterns, design patterns and idioms, which refer to the development stages of rough architecture, detailed architecture and implementation.

## 2.2 Patterns in Learning

The use of patterns in learning refers first of all to best practices for designing learning processes. Here, it is not the result of the process that is the aim of the pattern, but the process itself. In [13] and [14], for example, Köppe describes teaching approaches specifically for the teaching of design patterns, how this concept can be better understood by students and implemented in practical exercises. Warren describes active learning techniques for similar learning content in [15].

Many software applications are also being developed to support electronic learning. How such applications are structured and which learning principles are implemented is part of current research. For example, Fioravanti and Barbosa present a catalogue of 312 pedagogical patterns for software applications identified in 51 studies in [16].

Barros et al. propose a software tool that assists students when learning to program by using different patterns that are stored as sample code. This enables the learner to find quick solutions to minor implementation issues (such as choosing a suitable loop construction) [17]. Porter and Calder formalize the solution of programming tasks by a learner by gradually selecting patterns that refine the program structure and ultimately result in a runnable program [18].

Kolfschoten et al. were even able to show that the use of design patterns in design exercises not only increases the efficiency of task processing, but also enables more efficient learning of design knowledge for beginners [19].

### 3 PATTERN LANGUAGE FOR ACADEMIC WRITING

A modelling language for scientific texts is presented in the sections that follow. The aim is to build a pattern catalogue to represent texts structures on different content levels before the actual content of the texts is written. A thorough analysis of different text books, as well as the experience of the authors of this paper in the supervision of theses, determined both the selection and design of the patterns that will be introduced next. Finally, the patterns can be put together in such a way that they can be assigned to the writing phases and, as a pattern language, cover the entire writing process.

#### 3.1 Text Modelling Language

Just as the design language UML (Unified Modelling Language [19]) for software systems describes different perspectives of a system, it is necessary for texts to allow different levels of description. This becomes clear from the distinction between the conversion of a report into continuous text and graphics compared to rough planning based on an outline.

The text model language TML described by Faustmann [20] contains four different perspectives on scientific texts:

1. The **outline perspective** describes the well-known division of a text into main and sub-chapters. The presentation is done by numbering the text with the respective text titles.
2. The **function perspective** shows the functions of the individual text parts contained in the text. Functions include descriptive parts (DOCUM), analysing parts (ANALYS), comparative parts (COMP), hypotheses (HYPOTH), evidence presentation (EVID), evaluations (EVAL) and summaries (SUMM).
3. The **content perspective** contains brief descriptions of the content of the respective text parts.
4. The **presentation perspective** shows how each part of the text is presented to the reader. These forms of presentation include texts, enumerations or lists, tables, graphics, program code and mathematical expressions.

The content of the respective perspectives can be combined into one perspective, but also across perspectives. The following relationship types are used for this:

- **Sequence relationships** simply show the sequence of parts in the text. They can also be dropped if the sequence is already clear in the model. Sequence relationships are typically only modelled within one perspective. They are represented by a line with an open arrowhead.
- The **containment relationship** shows which elements are implemented by others. For example, evidence (EVID) for a hypothesis (HYPOTH) can be realized by a comparison (COMP). Containment relationships can be used within a perspective, but are generally cross-perspective. They are represented by a simple connecting line.
- The **reference relationship** clarifies the relationship between text parts. The reference of an evidence (EVID) to a hypothesis (HYPOTH) previously explained in the text should be displayed here. Thus, reference relationships often lead back in the text. These references are shown as dotted lines with an unfilled arrowhead.

#### 3.2 Text Patterns

As a result of the analysis of different textbooks about academic writing and our own experiences, ten patterns for scientific texts were initially identified, which form the basis for the desired pattern language. Due to space reasons, only three patterns are presented here in detail, followed by an overview of the remaining seven text patterns. The patterns presented in detail are described by the attributes shown in Table 2.

Table 2. Attributes that describe text patterns

Attribute	Meaning
Name	According to its use, each pattern contains a meaningful name
Purpose	An explanation of what can be achieved with the pattern in the text, or what a text would lack without this approach
Motivation	The advantage of using the pattern will be shown by means of an example
Structure	A text model is used to show which perspectives are affected by a text and which parts of the text have an influence
Consequences	Here the advantages and disadvantages from the point of view of both the reader and the author are pointed out
Implementation	Since texts are the target objects, references to concrete formulations or design aspects are given
Sources	Where do the rules that implement the pattern come from? The information should be as precise as possible (e.g. with page numbers) to allow for easy reference.

The additional attributes introduced in [8] for the description of software patterns (see Tab. 1) are not used here, because they partly do not fit the objective "text" (e.g. "known use"). The attributes "category" and "related patterns" are used in the following section where a text pattern language is developed. This leaves nine attributes, which are used to describe text patterns in the following sections (the "Name" attribute is already defined by the respective heading).

### 3.2.1 Hourglass Pattern

**Purpose:** The aim is to awake the interest of a broader readership for the issue under consideration. Without this structure, the text would only be of interest to a few experts.

**Motivation:** The question examined in the text is introduced by a broad context and developed more and more closely to the actual question. For example, the question of how to model texts is not very interesting for most readers. However, if examples of applications that are embedded in a broader field such as e-learning are shown, then the circle of readers could become considerably larger.

**Structure:** Figure 1 shows the Hourglass Pattern as a text model. It depicts the beginning and the end of the text (Introduction and Discussion). Both the narrowing and finally the broadening of the question takes place.

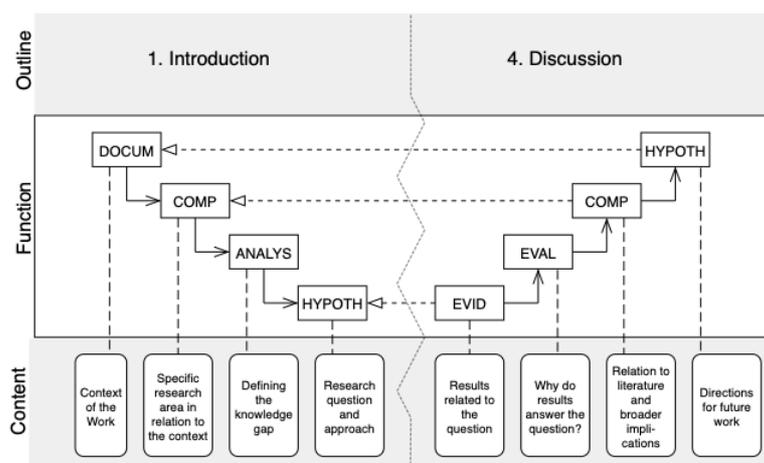


Figure 1. Structure of Hourglass Pattern

**Consequences:** For the reader, the question dealt with is better placed in a context that may be more interesting for him/her. However, this also means that the work only reaches the actual point (the analyzed question) later. The author of the text has to consider that it depends on the readership how broadly the context is chosen. In very specific journals, it is not necessary to describe a broad context

and is rather obstructive. The overall structure must be based on the IMRaD concept (Introduction, Methods, Results, Discussion), which provides for introduction and discussion.

**Implementation:** The implementation can be carried out as pure continuous text. The differentiation of the content parts should be done by separate paragraphs.

**Sources:** Heard, *The Scientist's Guide to Writing*, p.77ff [5]; Katz, *From Research to Manuscript*, p. 13ff [21]; Schimel, *Writing Science*, p. 33, 45 [22]

### 3.2.2 Complex Research Argument Pattern

**Purpose:** A question is first answered by a hypothesis (claim). This claim must be substantiated in a comprehensible way. This is done by giving reasons and analyzing substantiated statements (e.g. empirical studies).

**Motivation:** A scientific text that deals with a question must be based on this argumentation structure. The assumption that violent films influence the behavior of children in their later life (HYPOTH), as the boundaries between reality and fantasy become blurred through feature films (DOCUM for reason), must be secured (e.g. through studies investigating perception after watching a film - ANALYS).

**Structure:** Figure 2 shows the structure of the Complex Research Argument Pattern. While in the introduction the hypothesis is developed from the research question, in the discussion arguments for the hypothesis should be found. These are composed of reasons for the hypothesis and corresponding proofs that refer to the results of the work.

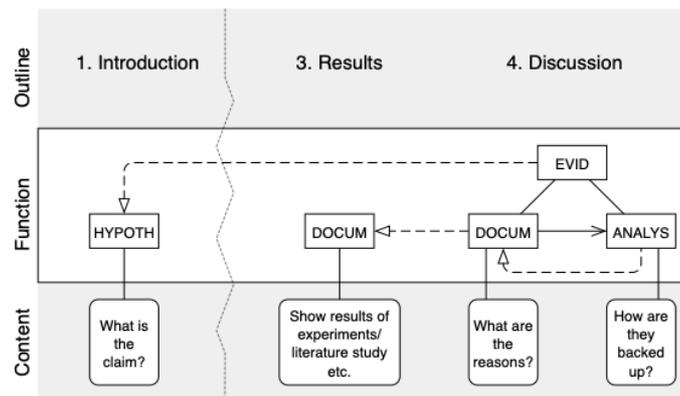


Figure 2. Structure of Complex Research Argument Pattern

**Consequences:** For the reader, this structure creates a comprehensible justification of the assertion, which is at best based on facts. However, the interpretation of these facts still leaves room for argumentative discussions. These counter-arguments should also be anticipated by the author, which can lead to a continuous string of reasons and evidence. It is up to the author to decide when an assertion is sufficiently justified.

**Implementation:** In empirical studies, evidence can often be presented through statistical analyses and resulting graphical evaluations (e.g. correlations). Graphical representations are also helpful for the proof of optimized structures, which e.g. show a better overview and thus a lower complexity of a system.

**Sources:** Turabian: *A Manual for Writers of Research Papers, Theses, and Dissertations*, p. 48ff [23]; Booth et al.: *The Craft of Research*, p.114ff [2], Toulmin: *The Uses of Argument*, p. 87ff [3]

### 3.2.3 Presentation Referencing Pattern

**Purpose:** For the reader, a text with its additional forms of presentation (such as graphics, tables, etc.) should offer an understandable reading flow. Presentations in the text that are not pure text should therefore be appropriately referenced by the written parts.

**Motivation:** Graphic representations support the reader's understanding of an analysis and generate interest by changing the media presentation. However, if a graphical data analysis is not directly related to the text, it can be difficult for the reader to understand the meaning behind the analysis.

**Structure:** Figure 3 shows the affected perspectives of a text for the Presentation Referencing Pattern. If the content includes graphics, tables, etc., a connection to the text parts must exist via a corresponding reference. This connection of content and different forms of presentation can appear in any part of the text.

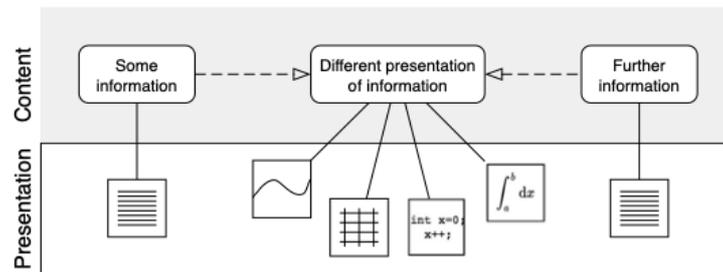


Figure 3. Structure of Presentation Reference Pattern

**Consequences:** This raises the question of how strong the connection between, for example, graphic representation and text must actually be. The author has to decide when to reference a graphic (before or after it is displayed in the text) and how exactly the text refers to the graphic. A description that is too precise can make the graphic itself superfluous, and a reference to the representation that is unclear in terms of content can be confusing for the reader or interrupt the flow of thoughts and reading.

**Implementation:** References in the text can be obtained by directly specifying the figure/table title including the number. Mathematical expressions can also be numbered and then referenced in the text.

**Sources:** Heard, *The Scientist's Guide to Writing*, p.102, 116/117 [5]

### 3.2.4 Further Patterns

This section briefly introduces other kinds of patterns that could be present in scientific writing.

- **Story-for-Explanation Pattern:** On the one hand, the pattern describes the basic IMRaD structure (Introduction - Methods - Results - Discussion) and, on the other hand, it specifies questions of content that are to be clarified in the respective chapters of the outline. Due to the nature of the questions, the pattern refers in particular to investigations as they occur in the natural sciences and attempt to explain certain facts.
- **Story-for-Design Pattern:** Like the Story-for-Explanation Pattern, the Story-for-Design Pattern describes the basic structure of a scientific paper, but here the focus is on the development of a prototype to solve certain problems (e.g. a new software tool). The main chapters are adapted in their titles and the questions revolve around the application problem, already known solutions, and the synthesis towards a new prototype solution.
- **Hypothesis Generation Pattern:** A scientific paper has to ask a research question and should come up with a hypothesis to answer this question. Using an example that illustrates the problem and an analysis of the research area, the question can be worked out and a first hypothesis developed.
- **Research Argument Pattern:** This pattern is a generalization of the Complex Research Argument Pattern described in the previous chapter. Here, forms of presentation are considered which support the various reasons in list form, especially for evidence.
- **Unified Paragraph Pattern:** The last three patterns refer to the structure of individual paragraphs. The Unified Paragraph Pattern requires a topic sentence for each paragraph, which outlines the goal of the paragraph. In the remaining paragraph, details and examples should be developed for this topic.
- **Distinct Paragraph Pattern:** In this pattern it is required that several paragraphs can be distinguished by their content (i.e. each paragraph describes a separate topic), but also that they must refer to each other by references. The references can be forward (e.g. as a conclusion of a paragraph), but also backward (e.g. in relation to a previous topic).

- **Coherent Paragraph Pattern:** This pattern requires a paragraph to arrange its contents according to a certain organization scheme. This organization can be time oriented (older to new events) or based on the importance of the arguments (less important to very important).

### 3.3 Text Pattern Language

The patterns described here must now be made available for a writing process. By assigning the patterns to the basic categories general structure, argumentation structure and text implementation, one has at the same time achieved an assignment to the phases of a writing process: In the first phase one will think about the rough structure of a text, which then leads to a first outline. This outline can already be developed further. At this point, however, it should already be clear what function the respective outline points will have. This then represents the storyline of the text for the reader. In the second phase, the arguments for the research hypothesis are determined and appropriately described. At this point it is therefore a question of the argumentation structure and the interaction with the previously defined outline. In the last phase the texts are created, which then have to convert the arguments into continuous text or suitable presentations. This is the implementation of the text. Figure 4 shows the described text patterns with the assigned categories.

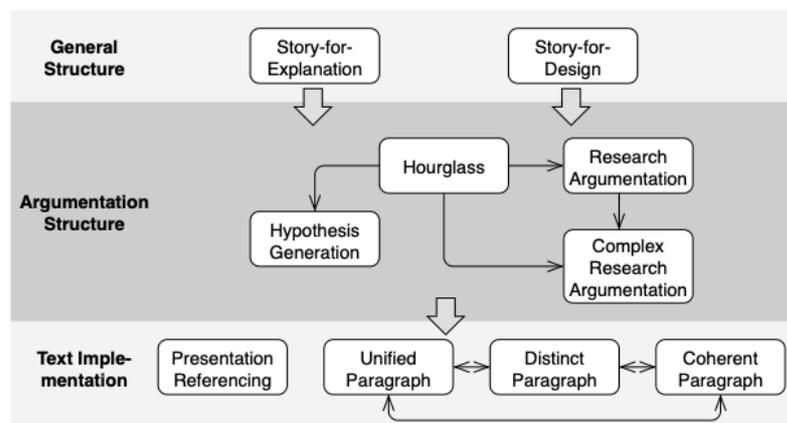


Figure 4. Text Pattern Language

The dependencies of the patterns, represented by connecting arrows in Figure 4, describe the procedure for using the patterns in the respective phases. While the Story-for-Explanation and Story-for-Design patterns can be chosen alternatively, the Hourglass Pattern is an essential part of the argumentation structure. In the text implementation, all the mentioned patterns should be used. The dependencies of the paragraph patterns refer to the integration of the three rules in one paragraph each.

## 4 CONCLUSIONS

### 4.1 Advantages

The proposed methodology for presenting rules for the creation of scientific texts using text models and sample descriptions leads to a compact and clear type of instructions for scientific writing. This is particularly positive in view of the fact that the literature on the subject is very extensive and students are unable to find their way around due to the large amount of advice. The result is often qualitatively unsatisfactory work.

Such a sample catalogue can significantly improve the process of writing in two ways: On the one hand, texts can be drafted more quickly, since the basic structures are available almost immediately and contain instructions on how to continue with the text parts even more detailed regarding the writer's own topic. On the other hand, the support process in the preparation of academic theses becomes more transparent, since students and supervisors can now use further indications for a thesis that go far beyond the previous outline points. Consequently, communication can also be adapted to the respective level of resolution of a text (rough structure, argumentation, text design).

## 4.2 Limitations

There are also disadvantages that the use of a pattern language can bring up. First, it requires a certain amount of effort to become familiar with the TML notation for text. This is comparable to other notations like UML for software engineering. Supporting means such as textbooks and tools for using TML can reduce this effort.

For a practical use of the presented pattern catalogue, a further development of patterns is necessary. The patterns described in this paper can be regarded as examples of how patterns look like and how they should be organized in a corresponding catalogue. Therefore, it would also be necessary to consider how the process of developing new patterns can be improved and made more efficient.

## 5 FUTURE WORK

From the disadvantages of the concept of a text pattern language described above, for example, it arises the requirement to specify further patterns and document them in a suitable form. Here, a textbook form that can be made available to students is a suitable option. This would make patterns practically applicable and generally documented available to students.

The practicability of the concept also depends on the availability of software tools that support both the process of creating patterns and their use in real writing processes. The following requirements for tool support should be mentioned here:

- Patterns should be available electronically in such a way that, in the case of a large number of patterns, they can be searched for appropriately (e.g. categories relating to a writing phase).
- This also includes the creation of new patterns on an appropriate platform (in principle a development environment for text patterns).
- For the user of the patterns, the question arises of how they can be integrated into the text creation process. Text processing systems are used for this purpose, although there are already more advanced solutions to support the text creation process by providing additional information (e.g. in the software Scrivener any information can be stored and used later when writing, [www.literatureandlatte.com/scrivener](http://www.literatureandlatte.com/scrivener)).
- As already introduced, there are different dependencies of patterns, which can also lead to a structural combination (e.g. hypothesis generation with research argumentation). A tool should support the composition of patterns in a text model that is created by the author.

Another theoretical point that can lead to an improvement of the concept is the modelling of anti-patterns, which are common mistakes in scientific papers. Examples are:

- the use of a theory or concepts of a theory, without introducing them properly,
- the formulation of research questions without scientific background or introduction to a problem or gap or missing research,
- the presentation of evidence, data, or supporting information for a problem that has not been described in advance,
- drawing conclusions without a previous discussion of the results that were obtained,
- addressing a problem that needs to be solved without describing it,
- presenting solutions to a problem without introducing the problem or the theory behind it,
- not giving supporting evidence to an argument for or against a hypothesis.

As it can be seen from these examples, the dependencies of the text parts are complex and often cross-perspective. However, models of these errors in the texts could help to avoid them.

## REFERENCES

- [1] E. B. White and W. I. Strunk, *The Elements of Style*, New York: Pearson Longman, 2010.
- [2] W. C. Booth, G. G. Colomb, and J. M. Williams, *The craft of research*. Chicago, Ill. [u.a.]: Univ. of Chicago Press, 2003.

- [3] S. E. Toulmin, *The Uses of Argument*, Updated edition. Cambridge, New York: Cambridge University Press, 2003.
- [4] S. Evergreen, *Presenting Data Effectively*, Los Angeles: SAGE Publications, 2017.
- [5] S. B. Heard, *The Scientist's Guide to Writing: How to Write More Easily and Effectively throughout Your Scientific Career*. Princeton Oxford: Princeton University Press, 2016.
- [6] P. Goodson, *Becoming an Academic Writer: 50 Exercises for Paced, Productive, and Powerful Writing*, Los Angeles: SAGE Publications, 2016.
- [7] P. J. Silvia, *How to Write a Lot: A Practical Guide to Productive Academic Writing*, Washington, DC: Amer Psychological Assn, 2018.
- [8] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software.*, 1st ed., Reprint. Reading, Mass: Prentice Hall, 1994.
- [9] C. Alexander, H. Neis, and A. Anninou, *A New Theory of Urban Design*. New York: Oxford University Press, 1987.
- [10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture, A System of Patterns*, 1st ed. Wiley, 2000.
- [11] T. Noll, *The Joint Book: The Complete Guide to Wood Joinery*, London: Chartwell Books, 2009.
- [12] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977.
- [13] C. Köppe, "A pattern language for teaching design patterns (part 1)," in *Proceedings of the 16th European Conference on Pattern Languages of Programs - EuroPLoP '11*, Irsee, Germany, 2011, pp. 1–21, doi: 10.1145/2396716.2396718.
- [14] C. Köppe, "A Pattern Language for Teaching Design Patterns (Part 2)," in *Proceedings of the 18th Conference on Pattern Languages of Programs*, New York, NY, USA, 2011, doi: 10.1145/2578903.2579161.
- [15] I. Warren, "Teaching Patterns and Software Design," in *Proceedings of the 7th Australasian Conference on Computing Education - Volume 42*, AUS, 2005, pp. 39–49.
- [16] M. L. Fioravanti and E. F. Barbosa, "A Catalog of Pedagogical Patterns for Learning Applications," in *2018 IEEE Frontiers in Education Conference (FIE)*, San Jose, CA, USA, 2018, pp. 1–9, doi: 10.1109/FIE.2018.8658904.
- [17] L. N. de Barros, A. P. dos Santos Mota, K. V. Delgado, and P. M. Matsumoto, "A tool for programming learning with pedagogical patterns," in *Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange - eclipse '05*, San Diego, California, 2005, pp. 125–129, doi: 10.1145/1117696.1117722.
- [18] R. Porter and P. Calder, "A Pattern-Based Problem-Solving Process for Novice Programmers," in *Proceedings of the Fifth Australasian Conference on Computing Education - Volume 20*, AUS, 2003, pp. 231–238.
- [19] G. Kolfschoten, S. Lukosch, A. Verbraeck, E. Valentin, and G.-J. de Vreede, "Cognitive learning efficiency through the use of design patterns in teaching," *Computers & Education*, vol. 54, no. 3, pp. 652–660, Apr. 2010, doi: 10.1016/j.compedu.2009.09.028.
- [20] G. Faustmann, "Improved Learning of Academic Writing - Reducing Complexity by Modeling Academic Texts", 10th Intern. Conf. on Computer Supported Education, 2018, pp. 447–453.
- [21] M. J. Katz, *From Research to Manuscript: A Guide to Scientific Writing*, 2nd ed. 2009. New York: Springer, 2009.
- [22] J. Schimel, *Writing Science: How to Write Papers That Get Cited and Proposals That Get Funded*, Oxford, New York: Oxford University Press, 2011.
- [23] K. L. Turabian, *A Manual for Writers of Research Papers, Theses, and Dissertations, Seventh Edition: Chicago Style for Students and Researchers*, Chicago: Univ. of Chicago Press, 2007.