

TUDaThesis – Abschlussarbeiten im CD der TU Darmstadt

L^AT_EX using TU Darmstadt's Corporate Design

Bachelorarbeit im Studienbereich Computational Engineering von Marei Peischl

Tag der Einreichung: 2. Februar 2023

1. Gutachten: Gutachter 1
2. Gutachten: Gutachter 2
3. Gutachten: noch einer
4. Gutachten: falls das immernoch nicht reicht
Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Studienbereich
Computational Engineering
Institut
Arbeitsgruppe

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Marei Peischl, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 2. Februar 2023

M. Peischl

Inhaltsverzeichnis

1	Problem introduction	4
2	Introduction	5
3	Verwendung	7
3.1	Sprachanpassung	7
3.2	Übergabe der Titeldaten	8
3.3	Weitere Macros	11
3.4	Layout-Optionen mit Verstoß gegen das Corporate Design	12
3.5	Spezielle Optionen für Abschlussarbeiten	14
3.6	Erhöhter Zeilenabstand – Hinweise zum setspace-Paket	16

1 Introduction

This work will be dedicated to the problem, which is very common to the business. Nowadays almost every (if not every) company that's coming from such fields like B2B (business-to-business), B2C (business-to-consumer), E-commerce, service-based, healthcare, NPOs (Non-Profit Organizations), Financial institutions, Telecommunications, Retail, Real estate or Transportation [FIND LITERATURE] uses one or several CRM Systems (Customer Relationship Management System) to manage internal and external organizational relationships, which also includes storing (and protecting) all sorts of the data, including the data about its products, activities, customers and shareholders etc. This data then could be utilized to provide a broad source of applications within of the company. One of such possible applications is building prediction or suggestion models to better understand the behaviour of the customers, product on the market or pricing development.

While to solve those problems there exists a broad set of tools, they would be useless unless there's a valid data to get insights from. For it one would first have to collect the data over some span of time and to prepare it to use. And this very point of collecting and processing data is the standing ground for the whole application to work accordingly, since it will have to rely on this data through its whole life-time.

Let's imagine a company hosting various events (also online-events) during some period of time, while also gathering the data about the participants in form of some registration form, that also includes the name of the participant. The use case that will appear at some point using this data would of course include the validation of such data.

This brings us to the point, where there's only one possible way to do it properly - match this data from registration sheets to the data of the CRM System. While we can be (almost) sure about validity of the CRM data, there's no way to be sure, that data from registration sheets is of the needed quality. Unconscious mistakes such as misspelling or getting typing errors in the own name are often a very common in the data like this. But also conscious misspelling such as writing the name in a different way or even using a nick-name instead of real name might coexist it the same data. And lastly, the events might also include

participants from outside of the company, which makes the data not only chaotic, but also sparse in the sense of possible matches.

Of course this is only one example of this type of problems, there're many more, just name a few: matching product names made by company using CRM Data and product names, which have been sold by a retailer, written by hand by the sales workers; matching therapeutic area of a doctor from the CRM system to a therapeutic area collected by sales employees etc.

All this cases are similar in few ways:

- it's a matching problem including two (often very big) data sets, which leads to performance issues;
- the data consists of character strings without any context;
- the data contains proper names, which don't really have or should have semantic sense;
- the collected data includes mistakes and mismatches with regard to a valid data.

We will call this set of problems **name matching**.

Using this knowledge we now can define our goal:

- 1 define mapping, that matches a fuzzy name to its valid analogue,
- 2 create the architecture of the solution,
- 3 perform the matching task in polynomial time (???)

2 Verwendung

Die Klasse kann wie für Dokumentenklassen üblich eingebunden werden

```
\documentclass[thesis]{tudapub}
```

Die Option `thesis` wechselt hierbei in den Modus, der spezielle Features für Abschlussarbeiten freischaltet, die in diesem Dokument beschrieben werden.

Darüber hinaus lässt sich die Klasse verwenden wie die Standard-KOMA-Script-Klasse, auf der sie basiert. Voreingestellt ist hierbei `scrreprt`.

Allgemein bietet KOMA-Script viele Möglichkeiten zu Anpassungen. Wie in der `tudapub`-Demo-Datei beschrieben, können hier jedoch nicht alle erläutert werden, ein Blick in die offizielle Dokumentation ist daher häufig hilfreich [[scrguide](#)].

2.1 Sprachanpassung

Der Modus für Abschlussarbeiten setzt einige sprachabhängige Bezeichnungen. Teilweise ist Deutsch für diese Elemente als Hauptsprache vorgeschrieben (z. B. die Selbstständigkeitserklärung). Für die korrekte Verarbeitung wird daher ein Paket zur Sprachanpassung benötigt. TUDa-CI verwendet hierfür das `babel`-Paket.

Dies wird jedoch nicht automatisch geladen, da hierfür die Konfiguration der Sprachen bekannt sein müsste. Die Demo-Dateien für Abschlussarbeiten (`DEMO-TUDaThesis.tex`/`DEMO-TUDaPhD.tex`) laden hierfür die Konfiguration:

```
\usepackage[english, main=ngerman]{babel}
```

Diese ist für ein Dokument mit Deutsch als Hauptsprache und Englischen Elementen. Die Hauptsprache wird als Wert der Option `main=` übergeben. Das Laden von `ngerman` wird in den Fällen, in denen es von TUDa-CI benötigt wird, automatisch ausgelöst. Für eine bessere Übersichtlichkeit ist es dennoch hilfreich es dort aufzuführen.

Falls die Hauptsprache nicht Deutsch ist, wäre daher die folgende Konfiguration sinnvoll:

```
\usepackage[ngerman, main=<Hauptsprache>]{babel}
```

2.2 Übergabe der Titeldaten

Die Daten werden analog zur klassischen Titeleierzeugung mit `\maketitle` übergeben. Allerdings wurden die Felder um ein paar speziellere Daten erweitert. Sofern nicht anders angegeben, verfügen alle Makros über ein notwendiges Argument für die Datenübergabe, z. B.

```
\title{\LaTeX{} im Corporate Design der TU Darmstadt}
```

Es ist zu beachten, dass für die Erzeugung der Titelseite nach Übergabe aller Daten `\maketitle` aufgerufen werden muss.

Falls eine Layoutanpassung der Titelseite notwendig ist, gelten die in der TUDaPub-Dokumentation geschilderten Optionen. Dort finden sich auch Hinweise zur Platzierung von Sponsorenlogos. [**tudapub**]

`\title` Titel, wird in sehr großer Schrift im obersten Block der Titelseite platziert. Die Schriftgröße ist aufgrund der Häufigkeit für lange Titel kleiner gewählt als für andere Publikationen.

`\subtitle` Untertitel. Dieses Feld kann alternativ für eine Übersetzung genutzt werden.

`\author` Der Autor/dir Autoren. Mehre Autoren werden durch `\and` getrennt.

`\studentID` Matrikelnummer. Nach den Vorgaben des Templates ist diese Angabe immer optional.

`\birthplace` Geburtsort. Angabe ist bei Dissertationen notwendig.

`\reviewer` Gutachter. Mehrere Gutachter werden, wie Autoren durch `\and` getrennt. Die Nummerierung läuft von links nach rechts.

Änderung des Bezeichners

Die Änderung des Bezeichners ist über ein optionales Argument möglich:

```
\reviewer[Ersatzbezeichner]{Name1 \and Name2}
```

Um die numerische Benennung abzuändern ist es zusätzlich möglich statt dem Ersatzbezeichner eine Kommaliste zu übergeben:

```
\reviewer*[Bezeichner1, Bezeichner2]{Name1 \and Name2}
```

In diesem Fall entfällt die Nummerierung vor dem Bezeichner. Soll z. B. den Formulierungen der Promotionsordnung entsprochen werden, gilt:

```
\reviewer[Erstreferent\_in, Koreferent\_in]{Name1 \and Name2}
```

Für die Erstellung fachbereichsspezifischer Templates existiert hierfür auch ein Makro, das ohne den Aufruf von `\reviewer` Änderungen zulässt.

```
\setupReviewName{Ersatzwort für „Gutachten“}
```

Setzt die ersten beiden Bezeichner. Alternativ ist es auch möglich Positionen einzeln zu benennen `\setupReviewName[1]{Erstferent}`, eine Übergabe als Kommaliste ist als `\setupReviewName*{Bezeichner1, Bezeichner2}` möglich.

`\institution` Einrichtung. Dieser Eintrag, wie auch die beiden folgenden, werden unterhalb des Logos auf der Titelseite platziert.

`\department` Fach-/Studienbereich, allerdings ist die oben genannte Option zu bevorzugen. Die Verarbeitung des Arguments erfolgt jedoch analog.

Dieses Makro verfügt jedoch zusätzlich über die Möglichkeit abweichende Einträge gegenüber den Vorgaben anzugeben. Insbesondere wenn eine gesonderte Formulierung gegenüber der voreingestellten „im Fachbereich“ und ihren Varianten notwendig ist. Hierfür liefert `\department` ein optionales Argument:

```
\department[Ersatztext]{Kürzel/Bezeichnung}
```

Zusätzlich gibt es ab Version 2.01 auch die Möglichkeit den gesamten Text „im Fachbereich <Bereichsbezeichnung>“, sowie die Angabe in der Infobox auf der Titelseite zu ersetzen. Dies geschieht über die gesternete Variante:

```
\department*[Text für die Box]{Text zwischen Typ und Autor}
```

`\group` Arbeitsgruppe.

`\submissiondate` Datum der Einreichung

`\examdate` Datum der Disputation

`\date` Beliebiges Datum. Wird über `datename` bezeichnet.

`\publishers` Wird hier für die Ortsangabe verwendet und ist mit „Darmstadt“, bzw. „Darmstadt, Technische Universität Darmstadt“ (bei Dissertationen) vorbelegt.

`\tuprints` Übergabe der Daten, sofern das dokument über `tuprints` Veröffentlicht werden soll.

```
\tuprints{
  urn=1234,
  printid=12345,
}
```

Falls das Argument kein Gleichheitszeichen erkennt, wird der Wert als `printid` gesetzt und keine URN angegeben.

Ab Version 2.07 ist es zudem möglich einen eigenen Lizenztext über den Schlüssel `license` zu übergeben. Dieser ersetzt dann die voreingestellte Lizenzangabe.

Ab Version 3.08 existieren vorgefertigte Werte für die Option `license`, um eine einfachere Anpassung zu ermöglichen. Dies ist Bestandteil der Vorbereitung zur Anpassung der Standardlizenz, die durch die Universitätsbibliothek angepasst wurde. Die entsprechende Diskussion findet sich unter https://github.com/tudace/tuda_latex_templates/issues/251 und eine Anpassung der Voreinstellung bei TUDa-CI ist für Version 4.0 vorgesehen.

Die vorgefertigten Lizenzschlüssel lauten: `cc-by-nc-nd-2.0-de` (noch aktuelle Voreinstellung), `cc-by-4.0`, `cc-by-sa-4.0`, `cc-by-nc-sa-4.0`, `cc-by-nc-4.0`, `cc-by-nd-4.0`, `cc-by-nc-nd-4.0`. Unterstützung bei der Wahl einer passenden Creative Commons Lizenz bietet das Projekt über seinen Lizenzfinder unter <http://creativecommons.org/choose/>. Ab Version 3.10 werden alle (Stand: Januar 2021) dort vorgeschlagenen Lizenzen von `tuda-ci` unterstützt.

Falls ein davon abweichender Wert gewählt wird, wird dieser direkt an der Stelle des Lizenztextes verwendet. Falls der Text Gleichheitszeichen oder Kommata enthält ist eine Gruppierung notwendig.

`\titlegraphic` Hier kann Code übergeben werden, der den farbigen Block im unteren Teil der Titelseite ersetzt. Details sind in der allgemeinen TUDaPub-Dokumentation beschrieben [**tudapub**]

`\titleintro` Ab Version 2.03 kann zusätzlich über diesen Hook ein beliebiger Text direkt nach dem Untertitel und vor den automatischen Daten ergänzt werden.

`\titleaddendum` Wie `\titleintro` jedoch als letztes Element des Blocks.

2.3 Weitere Macros

Das Makro `\affidavit` erzeugt eine Selbstständigkeitserklärung mit Unterschriftenzeile. Hier wird der oben übergebene Name/Signatur eingefügt. In diesem Dokument findet sich das Affidavit direkt nach der Titelei.

Ab Version 3.06 verfügt `\affidavit` zusätzlich über ein optionales Argument, über das der Modus eingestellt werden kann. Es besteht die Unterscheidung zwischen `digital` und `print`. Hier wird entsprechend der Vorgabe unter https://www.tu-darmstadt.de/studieren/studierende_tu/studienorganisation_und_tucan/hilfe_und_faq/artikel_details_de_en_37824.de.jsp der entsprechende Text automatisch ausgewählt.

`\affidavit[digital]`

Da für Dissertationen bisher keine Option der rein digitalen Abgabe besteht, entfällt dort diese Unterscheidungsmöglichkeit, hierfür wird intern der Schlüssel `dr` verwendet.

Version 3.20 ermöglicht zusätzlich die Übergabe weiterer Optionen für den Signatur-Namen, ein Signatur-Bild oder die Ortsangabe. Inwieweit diese Optionen verwendet werden dürfen ist jeweils vor der Verwendung durch die Nutzer:in abzuklären. TUDa-CI kann hierfür keine gesicherte Aussage treffen. In diesem Fall muss der Varianten-Option der Optionsschlüssel `affidavit` vorangestellt werden.

`\affidavit[affidavit=digital,signature=Signaturname,signature-image={\include`

Eine vertikale Verschiebung des Signaturbildes ist nicht direkt implementiert, ist jedoch mit der Verwendung des L^AT_EX-Makros `\raisebox{<Verschiebung>}{<Inhalt>}` problemlos möglich.

Es besteht zusätzlich die Möglichkeit ein anderssprachiges Affidavit als Ergänzung mit abdrucke. Um die Struktur und die ggf. notwendige Sprachumschaltung zu erledigen, existiert hierfür ab Version 2.03 eine Umgebung:

```
\begin{affidavit*}[Babel-Sprachoption]{Überschrift}
Text
\end{affidavit*}
```

Diese Variante verfügt bewusst über keine Unterschriftenzeile, da diese Version laut Verständnis der Entwickler keine rechtliche Verbindlichkeit besitzt.

Die Umgebung kann jedoch auch für besondere Formen der Erklärung genutzt werden. In diesem Fall kann eine zusätzliche Signaturzeile über

```
\AffidavitSignature[Stadt]
```

hinzugefügt werden. Die Vorbelegung für Stadt ist hierbei „Darmstadt“. Ab Version 3.20 ist die Übergabe einer zusätzlichen Option für den Ort der Signatur auch als Option möglich.

```
\affidavit[signature-location=Stadt]
```

2.4 Layout-Optionen mit Verstoß gegen das Corporate Design

Die Zeilenlängen sind laut Corporate Design aus typografischer Sicht zu lang. Daher existiert die Klassenoption `custommargins`, die für dieses Dokument aktiviert wurde (Wert `true`). Sie verfügt über die Werte `true`, `false` und `geometry` mit folgender Bedeutung:

custommargins=false Standardeinstellung von tudapub. Die Ränder entsprechen den Vorgaben des Corporate Design Guidelines. Die Einstellung wird durch `geometry` durchgeführt. Eigene Anpassungen werden durch das Ausführen von `\maketitle` überschrieben.

custommargins=true Die Einstellungen des Corporate Design Guidelines werden nicht aktiviert. `geometry` wird nicht geladen. Dieser Modus entspricht der Standardeinstellung von KOMA-Script. Dadurch werden die Ränder nicht fest definiert, sondern auf Basis des `typearea`-Paketes berechnet [[scrguide](#)].

custommargins=geometry Diese Variante wurde auf Wunsch zur Verfügung gestellt, allerdings wird darauf hingewiesen, dass manuelle Randeinstellungen oft nicht zu einem harmonischen Satzspiegel führt. `geometry` wird, wie bei `false` geladen und vorkonfiguriert. Es ist allerdings möglich kleinere Anpassung durch die Verwendung des Makros `\geometry` zu setzen. Die Einstellungen, die zu Beginn des Dokuments gelten werden gespeichert und nach der Titelseite wiederhergestellt.

Hierbei ist zu beachten, dass die Einstellungen als Ausgangspunkt den Voreingestellten Satzspiegel nutzen (je nach Option mit Randnotizspalte oder ohne). Es ist möglich diese Optionen vor den eigenen mit zurückzusetzen:

```
\geometry{
  reset,
  <Eigene Anpassungen>
}
```

Die gilt insbesondere für die Optionen `includehead`, `includefoot` und `includemp`.

Hinweis zu den Kopf-/Fußzeilen

Wenn die Option `marginpar=true` gesetzt bleibt, ragen die Kopf- und Fußzeile über die Marginalspalte hinaus. Aus ästhetischen Gründen wird daher empfohlen in diesem Fall die Kopf- und Fußzeile mit `marginpar=false` auf den Textbereich zu beschränken.

Auch ist das Standard-Layout der Kolumnentitel wenig vorteilhaft, da die Kolumnentitel damit local größer sein können als die eigentliche Überschrift. (`headline=automark`)
Deswegen kann über

```
\pagestyle{TUDa.headings}
```

ein einfacherer Seitenstil ausgewählt werden, der die Nutzung mit lebenden Kolumnentitel erheblich vereinfacht. Dieser Stil ist über `scrlayer-scrpage` realisiert und kann entsprechend der KOMA-Script-Dokumentation angepasst werden.

Hinweis zur Bindekorrektur

Bei Verwendung einer Bindekorrektur (BCOR=<Länge>) wird diese nicht automatisch auch auf der Titelseite eingefügt. Für diesen Fall wurde mit Version 3.0 zusätzlich die Option BCORtitlepage hinzugefügt. Falls diese aktiviert wird, nimmt die Titelseite den Wert der Typearea Option BCOR auf der ersten Seite als Zusatz zum linken Rand hinzu.

2.5 Spezielle Optionen für Abschlussarbeiten

Die Klasse unterstützt alle Optionen der tudapub-Klasse. Darüber hinaus besteht über Wertzuweisung der Option thesis die Möglichkeit spezielle Einstellungen zu wählen. Es ist prinzipiell möglich die Optionen auch direkt als Optionen zur tudapub-Klasse zu übergeben, allerdings ist dies aufgrund der schlechteren Übersicht nicht zu empfehlen.

Für dieses Dokument wurden beispielsweise die Optionen als

```
thesis={type=drfinal,dr=phil}
```

übergeben.

Im folgenden findet sich die Bedeutung der einzelnen Optionen:

type=<Wert> Auswahl des Typus. Dieser wird auf die Titelseite gesetzt und wählt zudem aus welche Daten für die Titelseite zwingend übergeben werden müssen. Es stehen die folgenden Werte zur Verfügung (die Werte in Klammern sind die notwendigen Titeldaten):

- **sta**: Studienarbeit (title, author, date)
- **diplom**: Diplomarbeit (title, author, submissiondate, reviewer, department)
- **bachelor**: Bachelorarbeit (title, author, submissiondate, department, reviewer)
- **master**: Masterarbeit (title, author, submissiondate, department, reviewer)
- **pp**: Project-Proposal (title, author, date, department)
- **dr**: vorgelegte Dissertation (title, author, submissiondate, birthplace, department, reviewer)

-
- **drfinal**: genehmigte Dissertation (title, author, submissiondate, examdate, birthplace, department, reviewer)

Wird ein Typus angegeben, der nicht erkannt wird, so wird der Text direkt übergeben. Notwendige Titelfelder über den Titel hinaus gibt es in diesem Fall nicht.

dr=<Kürzel> Lädt einen der vordefinierten Texte für die Titelseite. Als Werte stehen bislang **rernat**, **ing** und **phil** zur Verfügung. Zum Beispiel lädt der Wert **phil**:

Zur Erlangung des Grades eines Doktor der Philosophie (Dr. phil.)

Sofern keiner dieser Werte dem angestrebten Titel entspricht, kann ein Text direkt übergeben werden.

`\drtext{Zur Erlangung des Grades \ldots}`

department=<Kürzel> Die Fachbereiche sind fest als Textbausteine in Deutscher sowie Englischer Sprache hinterlegt. Diese Option ermöglicht die Auswahl als Dokumentenklassenoption. Aus Kompatibilitätsgründen kann jedoch auch das Makro **department**-Makro hierfür genutzt werden. Zur Verfügung stehen:

arch	→ Architektur
bauing	→ Bau- und Umweltingenieurwissenschaften
bio	→ Biologie
chem	→ Chemie
etit	→ Elektrotechnik und Informationstechnik
gugw	→ Gesellschafts- und Geschichtswissenschaften
humanw	→ Humanwissenschaften
inf	→ Informatik
mb	→ Maschinenbau
matgeo	→ Material- und Geowissenschaften
math	→ Mathematik
phys	→ Physik
wi	→ Rechts- und Wirtschaftswissenschaften

Neben den Fachbereichen existieren für Abschlussarbeiten, die keine Dissertationen sind auch Studienbereiche. Falls das Kürzel nicht als Fachbereich hinterlegt ist, wird automatisch auf die Studienbereiche geprüft. Die Studienbereiche haben die folgenden Kürzel:

ce → Computational Engineering
ese → Energy Science and Engineering
ist → Information Systems Engineering
mech → Mechanik
metro → Mechatronik

Falls etwas anderes als eines dieser Kürzel übergeben wird, wird der Text direkt verwendet und eine entsprechende Warnung ausgegeben.

Die Auswahl der Fachrichtung erzeugt zusätzlich eine Box auf der Titelseite unterhalb des Logos. Falls diese automatische Erstellung nicht gewünscht ist, kann dies über die Option `instbox=false` deaktiviert werden.

ignore-missing-data Diese Option ist ein Schalter, der es ermöglicht die Fehlermeldung über nicht übergebene Titeldaten auszuschalten. In diesem Fall wird lediglich eine Warnung erzeugt, falls die angegebenen Daten nicht mit den Anforderungen übereinstimmen.


Abweichung von den Vorgaben für die Titelseite

Da es möglich sein kann von dieser Vorgabe abzuweichen, existiert für Sonderfälle die Dokumentenklassenoption `instbox=false`. Damit wird die automatische Verarbeitung der Daten für die Boxen auf der der Titelseite unterdrückt. In diesem Fall ist der Autor jedoch selbst für die Einhaltung der Vorschriften verantwortlich. Weitere Information zur Konstruktion der Boxen findet sich in den Verwendungshinweisen der Basisklasse TUDaPub. Zusätzlich sei auf die Möglichkeiten des `\department`-Makros verwiesen, sofern die Abweichung sich auf den Text beschränkt.

2.6 Erhöhter Zeilenabstand – Hinweise zum `setspace`-Paket

Sofern die Vorgaben es erfordern, ist es möglich mit dem `setspace`-Paket den Durchschuss zu erhöhen. Allerdings beeinflusst dies natürlich sämtliche Zeilenabstände. Ein erhöhter Zeilenabstand sollte daher erst nach der Titelseite aktiviert werden. Allgemein ist es jedoch empfehlenswert auch für Verzeichnisse und sonstige Sonderelemente außerhalb des Fließtextes auf bei normalen Einstellungen zu bleiben.

Setspace liefert hierfür die Möglichkeit, das Paket ohne Optionen zu laden und später über Makros, wie `\onehalfspacing` das Umschalten zu verzögern. Alternativ kann auch



durch die Umgebungen, wie `singlespace` lokal wieder zum Normalzustand gewechselt werden, sofern dies erforderlich ist.