
TUDaPub – L^AT_EX-Paper im Corporate Design der TU Darmstadt

Die Dokumentenklasse tudapub
Marei Peischl* und der T_EX-Löwe
18. Mai 2021

*peiT_EX T_EXnical Solutions



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Institutsbezeichnung bzw.
Logo in 2/3 Größe, vgl.
addTitleBoxLogo

Zusammenfassung

Dieses Dokument stellt ein Template und gleichzeitig die Verwendungsdocumentation zur Dokumentenklasse tudapub, einem Teil des TUDa-CI-Paketes dar.

Noch befindet sich das Projekt in der Testphase. Sollten Sie Fragen, Wünsche und Anregungen haben, bitten wir um eine entsprechende Mitteilung.

Abstract

Example for an additional abstract in English.

Inhaltsverzeichnis

1	Über diese Datei	2
2	Verwendung	2
2.1	Klassensoptionen	3
2.2	Die Titelseite	4
2.3	Strukturierungselemente	5
2.4	PDF/A Konformität	5
2.5	Zusätzliche Metadaten nach Wunsch der Universitätsbibliothek	6
2.6	Farbdarstellung	6
3	Erweiterte Konfigurationsmöglichkeiten	6
3.1	Auswahl des Dokumentenfarbmodus	6
3.2	Anpassungen, die von den Corporate Design Richtlinien abweichen	6
3.2.1	Schriftgröße	6
3.2.2	Seitenränder	7
3.3	Frontmatter/Mainmatter/Backmatter	7
3.4	Mathematikschriften	7
3.5	pdf _{La} T _E X	7
4	Fachbereichsspezifische Anpassungen	8
4.1	Fachbereich Maschinenbau	8
5	Weitere Konfigurationsmöglichkeiten: Standard-KOMA-Script	8
6	Bekannte Probleme	9
6.1	X _{La} T _E X und PDF/A	9
6.2	DVI-Ausgabe	9
6.3	Geschachtelte Akzente (Inkompatibilität zwischen pdfx und amsmath)	9
6.4	Möglicher Option Clash mit microtype	9

1 Über diese Datei

Die Datei DEMO-TUDaPub.tex beziehungsweise ihre Ausgabe DEMO-TUDaPub.pdf ist die Dokumentation der Dokumentenklasse tudapub.sty.

Sie ist Teil des TUDa-CI-Bundles und basiert in Teilen auf dem tudesign-Paket von C. v. Loewenich und J. Werner.

In diesem Dokument werden die speziellen Optionen und Einstellungsmöglichkeiten erläutert.

2 Verwendung

Die Klasse wird wie gewohnt geladen:

```
\documentclass[<Optionen>]{tudapub}
```

Im folgenden werden die möglichen Optionen beschrieben.

2.1 Klassenoptionen

class=<article|report|book> Diese Option legt die Basisdokumentenklasse fest. Die Werte laden die entsprechende KOMA-Script-Klasse [3]. Der Wert `article` lädt somit die Klasse `scrartcl`.

KOMA-Script ist eine Sammlung von Klassen und Paketen für \LaTeX , die neben den typografischen Anpassungen an den Europäischen Raum auch die Konfigurationsmöglichkeiten stark erweitert.

color=<Farbe> Wählt die Schmuckfarbe für die Nutzung in der Identitätsleiste aus. Die Farbcodes finden sich in der Farbübersicht in den Corporate Design Richtlinien. Neben diesen Farben kann prinzipiell jede beliebige Farbe übergeben werden. Die Optionen `accentcolor`, `textaccentcolor` und `identbarcolor` werden analog direkt an `tudacolors` übergeben. Auf diesem Weg können die Farben unabhängig voneinander gesetzt werden.

marginpar Schaltet die Randnotizspalte um. Voreingestellt ist `auto`. Dies bedeutet, dass die Randnotizspalte wie im Corporate Design Handbuch [1] über die fünfte Spalte läuft. . Darüber werden auch die Werte `true` und `false` akzeptiert. `false` setzt die Breite der Randnotizspalte auf 0. Der Mechanismus selbst wird nicht deaktiviert.

Randnotizen werden über den `komafont`-Mechanismus [vgl. 3] im Element `marginpar` gesetzt. Seine Voreinstellung entspricht

```
\setkomafont{marginpar}{\accentfont}
```

Um zusätzlich farbige Randnotizen zu setzen, könnte dies geändert werden, über

```
\addtokomafont{marginpar}{\color{textaccentcolor}}
```

twocolumn Aktiviert den zweispaltigen Modus global. In diesem Fall werden jedoch aufgrund ihrer Natur zwei Randnotizspalten erzeugt. Eine Nutzung in Kombination mit `marginpar=auto` ist daher in den meisten Fällen fragwürdig. Falls der zweispaltige Modus lediglich lokal aktiviert wird, entfällt dieses Verhalten, allerdings werden dann Randnotizen deaktiviert.

ruledheaders Wählt den Stil der Überschriften aus. `ruledheaders=all` wählt den mit Linien eingerahmten Stil für alle bis zur `\subsubsection`. Bei `chapter` beziehungsweise `section` ist dieser Stil entsprechend beschränkt. `false` lädt den Standardstil aus KOMA-Script.

type Als Typ stehen im Moment `publication` und `thesis` zur Verfügung. Die besonderen Möglichkeiten im Typ `thesis` sind in der Datei `DEMO-TUDaThesis.tex/.pdf` geschildert. Voreingestellt ist `publication`. Zusätzlich existiert ab Version 1.2 noch der Modus `intern`. Dieser wählt die Optionen `titlepage=false` für einen Titelpfopf statt Titelseiten, sowie die TUDaPub-Optionen `pdfa=false` und `IMRAD=false`. Dieser Modus ist für kurze, interne Berichte gedacht.

headline Die Kopfzeile verfügt über die im Corporate Design beschriebenen Layoutmöglichkeiten über den Wert `automark`. Da diese Lösung typografisch nicht sonderlich sinnvoll ist, ist es auch möglich, diese abzuschalten. Voreingestellt ist ein Stil ohne Kolumnentitel.

logo Option für die Titelseite, siehe 2.2

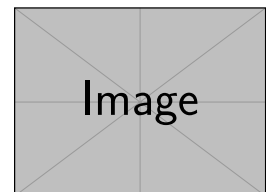
colorback Option für die Titelseite, siehe 2.2

IMRAD=true/false Deaktiviert die Prüfung auf IMRAD-Labels, siehe Abschnitt 2.5.

logfile=<Datei|pfad> Erlaubt es ein alternatives Logo zu übergeben. Diese Option existiert, um die Templates auch ohne das TUDa-Logo nutzen zu können. Die Logos sind der internen Verwendung vorbehalten und dürfen daher nicht mit diesem Template veröffentlicht werden.

Der Rest der Dokumentenklasse entspricht dem Standard von KOMA-Script, vgl. Abschnitt 5.

Beispiel für eine
Randnotiz



Flattersatz in der
marginline aus
KOMA-Script

2.2 Die Titelseite

Die Titelseite wird von tudapub automatisch generiert. Die Verwendung hierfür entspricht größtenteils der klassischen Methode unter Verwendung von `maketitle`.

Die KOMA-Script-Option `titlepage` erlaubt es üblicherweise, zwischen Titelseiten und Titelpopf umzuschalten. Bis Version 1.2 war diese Option deaktiviert. Mittlerweile existiert ein Modus für einen Titelpopf, dieser entspricht jedoch nicht den offiziellen Vorgaben und ist für interne Verwendung gedacht. Aufgrund der Implementierung wird `titlepage=true` identisch zu `titlepage=firstiscover` behandelt.

Die Makros wie `author`, `title`, `subtitle` und `date` entsprechen der normalen Verwendung. Für die Institutszuweisung kann ebenfalls wie gewohnt `thanks` verwendet werden.

Zusätzlich stehen die Makros `titleimage` und `addTitleBox` zur Verfügung um weitere Daten zu übergeben.

titleimage

Das Makro `titleimage` akzeptiert beliebigen Inhalt. Dieser wird bündig mit der oberen Ecke im Hauptteil der Titelseite platziert. Üblicherweise wird dieses Makro zur Platzierung einer Grafik genutzt:

```
\titleimage{\includegraphics[width=\width]{example-image}}
```

Zusätzlich stehen die Makros `titleimage` und `addTitleBox` zur Verfügung, um den unteren Teil der Titelseite zu füllen.

```
\titleimage{\includegraphics[height=\height]{titleimage}}
\addTitleBox{Institut 1}
```

Die Institutsboxen werden mit vorgegebenem Abstand unter dem Logo platziert. Hier kann Text erscheinen oder auch ein Institutslogo. Der Hintergrund ist weiß.

Um die Institutsboxen für Logos zu verwenden, liefert tudapub das Makro `addTitleBoxLogo`. Als Argument akzeptiert es einen Bilddateipfad.

```
\addTitleBoxLogo{example-image}
\addTitleBoxLogo*{\includegraphics[width=\linewidth]{example-image}}
```

Sponsorenlogos

Ab Version 3.0 steht in TUDaPub auch der Sponsorenmechanismus der TUDaLeaflet-Klasse zur Verfügung. Damit ist es möglich Sponsorenlogos unterhalb des Titelbildes zu platzieren.

Sponsorenlogos werden üblicherweise über

```
\AddSponsor{<logo1>}
\AddSponsor{<logo2>}
```

übergeben. Innerhalb des Arguments ist `\height` so gesetzt. Somit werden im Beispiel alle Logos auf die gleiche Höhe gesetzt. Der Abstand dazwischen wird entsprechend aufgefüllt, sodass der gesamte Block immer links und rechts mit dem Text abschließt.

Die zweite Variante ermöglicht die Platzierung mit manueller vertikaler Ausrichtung, wie es bei Logos mit unterschiedlicher Höhe notwendig sein könnte. Hierbei werden lediglich die Abstände und Trennlinien um die Logos ergänzt:

```
\sponsors{
<logo1><logo2>
}
```

Optionen für die Titelseite

Die Position des Logos ist umschaltbar. Dies geschieht über die Dokumentenklassenoption `logo=head/body`.

logo=head Das Logo wird im Kopf direkt neben dem Titel platziert. Der Titel wird in der Breite reduziert. Der Hintergrund des Titels wird in der Farbe der Identitätsleiste eingefärbt. Zusätzliche Infoboxen (s.u.) werden ebenfalls im Kopf platziert.

logo=body Das Logo samt der Infoboxen wird im Körper der Titelseite platziert.

Darüber hinaus lässt sich die Farbgebung umschalten. Die Option `colorback` schaltet zwischen dem farbigen Block auf der Titelseite und weißem Hintergrund um. Sie ist in der Voreinstellung aktiviert, sodass ein farbiger Block erzeugt wird.

Mit Version 3.08 werden zusätzlich zu den `true/false` Werten auch die Werte `title`, `head` und `body` für die Option `colorback` ergänzt. Mit diesen ist es möglich nach der Auswahl der Logoposition eine Korrektur an der Farbgebung durchzuführen. Alle drei neuen Werte aktivieren die Hintergrundfarbe und schalten deren Position um:

colorback=title Der Titel (ohne Untertitel) wird hinterlegt.

colorback=head Der komplette Titelblock, inklusive Titelblock wird hinterlegt.

colorback=body Es wird nur der Bildbereich eingefärbt.

2.3 Strukturierungselemente

Die abstract-Umgebung

Die `abstract`-Umgebung wird für `tudapub` um eine Option für die Sprache erweitert. Somit ist es möglich, mehrere Zusammenfassungen in einem Dokument zu nutzen.

```
\begin{abstract}
Zusammenfassung entsprechend der Dokumentensprache. In diesem Fall Deutsch.
\end{abstract}
```

```
\begin{abstract}[english]
Zusätzliche Zusammenfassung in englischer Sprache
\end{abstract}
```

Für die Verwendung ist wichtig, dass alle im Dokument genutzten Sprachen geladen werden. Im Falle des Beispiels muss also sowohl `ngerman` als auch `english` an das `babel`-Paket übergeben werden.

2.4 PDF/A Konformität

Die Klasse `TUDaPub` unterstützt den Standard PDF/A 2b. Der PDF/A-Modus ist automatisch aktiviert. Die zugehörige Option kann jedoch über `pdfa=false` ausgeschaltet werden. Nun wird zusätzlich eine `.xmpdata`-Datei generiert. Üblicherweise werden die Titeldaten direkt übernommen.

Dies kann jedoch bei der Verwendung einiger Makros innerhalb der Felder zu Problemen führen. Beispielsweise enthält der Titel für dieses Dokument das Makro `LATEX`. Es können daher nur Textelemente übernommen werden. Ähnlich den Linkbezeichnungen über PDF-Lesezeichen.

Um dieses Problem zu umgehen stellt `tudapub` hierfür das Makro `\Metadata{}` zur Verfügung. Hier können sämtliche von `pdfx` verarbeitbaren Variablen nach Schlüssel=Wert-Struktur gesetzt werden. Es ist zu beachten, dass dieses Makro nur dann funktioniert, wenn die `pdfa`-Ausgabe aktiviert ist. Ist dies nicht der Fall, so gibt `tudapub` eine entsprechende Fehlermeldung zu diesem Widerspruch aus. Zum Beispiel:

```
\Metadata{
author=Marei Peischl (peiTeX),
title=LaTeX im Corporate Design der TU Darmstadt,
}
```

Das Feld `publisher` ist mit „TU Darmstadt“ vorbelegt, kann aber überschrieben werden.

Um mehrere Einträge zu trennen, wird das Makro `\sep` genutzt.

```
keywords={TU Darmstadt \sep Corporate Design \sep LaTeX}
```

Wenn der Eintrag selbst Kommata enthalten könnte, dann ist eine Gruppierung um den Eintrag notwendig. Sonst wird der Text nach dem Komma als nächstes Keyword interpretiert.

Mögliche Probleme mit älteren Systemen:

Bei älteren TeX-Distributionen kann es vorkommen, dass die Farbprofile nicht vorinstalliert sind. In diesem Fall wird eine Fehlermeldung im folgenden Sinn erzeugt:

```
No color profile found to use for RGB screen colors
```

Um diesen Fehler zu beheben, können die notwendigen *.icc-Dateien unter <http://mirror.ctan.org/support/colorprofiles> heruntergeladen und entweder installiert oder im Projektordner mit abgelegt werden. Die einfachste Lösung bleibt jedoch, das eigene TeX-System zu aktualisieren.

Darüber hinaus werden bei XeTeX einige Features nicht unterstützt. In diesem Fall erzeugt `pdfa=false` ein kompilierbares Dokument, allerdings sollte für validierbare PDF/A-Dateien auf LuaTeX einer möglichst aktuellen Version umgestiegen werden.

2.5 Zusätzliche Metadaten nach Wunsch der Universitätsbibliothek

Es existiert ein Mechanismus um das Strukturierungsmodell IMRaD [2] zu kennzeichnen. Im Stil der einzelnen Teilbereiche können so, über Aufruf von

```
\IMRADlabel{introduction}  
\IMRADlabel{methods}  
\IMRADlabel{results}  
\IMRADlabel{discussion}
```

entsprechende Labels generiert werden. Sie haben den Namen `IMRAD:<Schlüssel>`.

Der Prüfmechanismus ist auf Wunsch der Bibliothek standardmäßig aktiviert, kann jedoch durch die Option `IMRAD=false` deaktiviert werden.

2.6 Farbdarstellung

Der PDF/A-Modus konvertiert automatisch eingegebene CMYK-Farben in RGB. Da es keine eindeutige Umwandlung gibt, sollten daher im Modus `pdfa=true` keine CMYK-Elemente verwendet werden.

Für die Druckausgabe ist dieser Modus ungeeignet. Ab Version 3.12 gibt tudapub eine entsprechende Infomeldung aus, um über diese Umwandlung zu informieren.

3 Erweiterte Konfigurationsmöglichkeiten

3.1 Auswahl des Dokumentenfarbmodus

Farben unterliegen je nachdem auf welchem Medium sie ausgegeben werden einer anderen Farbmischung. Für die Verwendung bei der Dokumentenerzeugung ist daher wichtig welches Ausgabemedium primär genutzt werden soll.

Technisch zeigt sich dieser Unterschied in Farbmodellen. TUDa-CI unterstützt entsprechend der Richtlinien sowohl ein Farbmodell für die Druckausgabe (cmyk) und für die Bildschirmdarstellung (RGB). Die Umsetzung geschieht über das Paket xcolor wobei die entsprechenden Farbwerte für beide Modelle über tudacolors hinterlegt sind.

Normalerweise wählt TUDa-CI automatisch ein passendes Modell. Allerdings sorgt die Voreinstellung von `pdfa=true` für eine Konvertierung zu RGB (vgl. 2.6).

Soll abseits dieser automatischen Abweichung ein bestimmter Farbmodus erzeugt werden, können die xcolor-Optionen cmyk oder RGB direkt an tudapub übergeben werden. Sie werden entsprechend an das Paket weitergereicht und entsprechend der xcolor-Dokumentation verarbeitet.

3.2 Anpassungen, die von den Corporate Design Richtlinien abweichen

3.2.1 Schriftgröße

tudapub kann entgegen der Corporate Design Richtlinien auch andere Schriftgrößen verarbeiten. Hierfür wird die `fontsize`-Option aus KOMA-Script genutzt (z. B. `fontsize=11pt`). Sofern keine spezielle Schriftgrößendatei für TUDa-CI vorliegt, wird die mit KOMA-Script ausgelieferte Datei gewählt.

Beispiele für Abweichungen aus typografischen Gründen sind Beispielsweise auch in den Demo-Dateien für Abschlussarbeiten gezeigt.

3.2.2 Seitenränder

Die Zeilenlängen sind laut Corporate Design aus typografischer Sicht zu lang. Daher existiert die Klassenoption `custommargins`, die für dieses Dokument aktiviert wurde.

Bei Verwendung einer Bindekorrektur 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.

Die Option `custommargins` verfügt ab Version 1.10 auch über den Wert `geometry`. Damit können die Ränder auch durch einen Aufruf von `\geometry` vor Beginn des Dokuments manuell angepasst werden.

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`.

Diese Variante wird auf Wunsch zur Verfügung gestellt, allerdings wird darauf hingewiesen, dass manuelle Randeinstellungen oft nicht zu einem harmonischen Satzspiegel führen.

Auch ist das Standard-Layout der Kolumnentitel wenig vorteilhaft, da die Kolumnentitel damit `local` größer sein können als die eigentliche Überschrift.

Dadurch werden die Ränder nicht fest definiert, sondern auf Basis des `typearea`-Paketes optimiert.

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.

Darüber hinaus kann über

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

ein einfacherer Seitenstil ausgewählt werden, der die Nutzung mit lebenden Kolumnentitel erheblich vereinfacht.

3.3 Frontmatter/Mainmatter/Backmatter

Üblicherweise existieren die Makros `\frontmatter`, `\mainmatter` und `\backmatter` lediglich bei der Basisklasse `scrbook`. Auf Wunsch wurden diese Makros auch bei `scrartcl` und `scrreprt` als Basis bereitgestellt.

Somit ist es möglich, für den Vorspann auf römische Ziffern zu wechseln. Ab `\mainmatter` werden dann arabische Ziffern verwendet.

3.4 Mathematikschriften

Da es keine Compiler-unabhängige, universelle Mathematikschrift gibt und die Corporate Design Richtlinien auch keinerlei Empfehlung berücksichtigen, wurden hierfür einige mögliche Varianten diskutiert. Die Voreinstellung entspricht immer dem Standard der Installation. Es werden keine spezifischen Einstellungen geladen.

Die Diskussion hierzu findet sich unter:

https://github.com/tudace/tuda_latex_templates/issues/19

Im Folgenden werden ein paar Beispielkonfigurationen gezeigt. Grundsätzlich ist die Mathematikschriftart jedoch – abgesehen durch Einschränkungen des Compilers – frei wählbar.

Bei Auswahl und Verwendung ist häufig der „ \TeX Font Catalogue hilfreich“: <https://tug.org/FontCatalogue/mathfonts.html>

3.5 pdf \LaTeX

Hier existiert eine Variante, die die Buchstaben der Basisschriftart „Charter“ mit Mathematiksymbolen aus unterschiedlichen Zeichensätzen möglichst passend kombiniert.

```
\usepackage[ charter , expert ] { mathdesign }
```

Es gibt ähnliche Ansätze für ein paar weitere Kombinationen. Einige Beispiele finden sich in der XCharter Dokumentation. <http://mirrors.ctan.org/fonts/xcharter/doc/xcharter-doc.pdf>

4 Fachbereichsspezifische Anpassungen

Einige Fachbereiche haben spezielle Anforderungen. Dieser Abschnitt betrachtet die speziellen Anpassungen. Bisher (Juli 2020) existieren diese Modifikationen lediglich für den Fachbereich Maschinenbau, der Mechanismus ist jedoch erweiterungsfähig.

4.1 Fachbereich Maschinenbau

Der entsprechende Modus wird über die Option `department=mecheng` oder kurz `mecheng` aktiviert. Die Farbgebung passt sich dann automatisch an und die Trennlinie der Fußzeile erhält die geforderte Form des Zeitstrahles.

Darüber hinaus erfordern manche Stellen dieses Fachbereichs die Übergabe einer ID zu Kennzeichnung des Dokuments. Hierfür wurde der Mechanismus

```
\SetPaperID{<Buchstabe>}{<tiefgestellte Nummer>}}
```

eingeführt. Dieser funktioniert auch ohne die Aktivierung des `mecheng`-Modus, allerdings ergänzt die Option einige zusätzliche Parameter für angepasste Abstände.

Darüber hinaus aktiviert der Modus die Optionen: `colorback=false`, `ruledheaders=section`.

Logo

Das Fachbereichslogo wird über die Option `departmentlogo=tuda_maschinenbau_logo` übergeben. Über diese Option kann auch eine abweichende Datei genutzt werden. Falls der Wert leer bleibt, wird kein Bild eingefügt.

Farbanpassungen

Der Fachbereich untergliedert die im Corporate Design Handbuch beschriebenen Farben zusätzlich. Daher existieren, wenn `mecheng` aktiviert wurde zusätzlich die folgenden Farbnamen:

```
\colorlet{TUDa-Primary1}{TUDa-6b}
\colorlet{TUDa-Primary2}{TUDa-2d}
\colorlet{TUDa-Secondary1}{TUDa-9a}
\colorlet{TUDa-Secondary2}{TUDa-8a}
\colorlet{TUDa-Secondary3}{TUDa-6a}
\colorlet{TUDa-Secondary4}{TUDa-3a}
\colorlet{TUDa-Secondary5}{TUDa-4a}
\colorlet{TUDa-Secondary6}{TUDa-5a}
\colorlet{TUDa-Arrow}{TUDa-Primary2}
```

Zeitstrahl

Des Design-Element des Zeitstrahls kann über das Makro `\MechEngArrow{<Länge>}` erzeugt werden. Die Farbe entspricht dabei der Farbe `TUDaArrow`, die mit der zweiten Primärfarbe (blau) vorbelegt ist.

5 Weitere Konfigurationsmöglichkeiten: Standard-KOMA-Script

Da die Klasse bis auf ein paar erzwungene Einstellungen, die das Layout betreffen, vollständig KOMA-Script-kompatibel ist, ist für sämtliche Modifikationen ein Blick in die KOMA-Script-Dokumentation hilfreich. Für einen Großteil bietet KOMA-Script eine eigene Lösung, wodurch Ergänzungspakete oft hinfällig sind.

Beispiele für typische Modifikationen, die auch im Rahmen des Corporate Design zulässig sind:

- Umstellung der Absatzkennzeichnungsmethode von Einzugs auf Abstand (Klassensoption `parskip`)
- Elementnummerierung mit oder ohne Endpunkt (Klassensoption `numbers=enddot/noenddot`)
- Positionierung, Ausrichtung und Abstände bei captions (Die Makros `\captionsabove`, `\captionbelow`, `\captionof` und die Option `captions`)

6 Bekannte Probleme

6.1 Xe_{La}TeX und PDF/A

Das Paket `pdfx`, über welches die PDF/A Kompatibilität erzeugt wird, hat nur einen begrenzten Support für Xe_{La}TeX. Es wird eine entsprechende Warnung erzeugt, allerdings kann es bei älteren Xe_{La}TeX-Versionen passieren, dass `pdfx` bereits Fehlermeldungen erzeugt. Abhilfe wird in diesem Fall durch einen Compiler-Wechsel auf Lua_{La}TeX (welcher ohnehin empfohlen wird) oder durch Abschalten des PDF/A-Modus (`pdfa=false`) geschaffen.

6.2 DVI-Ausgabe

Aufgrund der Voreinstellung zur Erzeugung einer PDF/A-Datei ist es nicht möglich TUDa-CI in Standard-einstellungen zur Erzeugung einer DVI-Datei zu nutzen. Ein Großteil der Funktionalität kann jedoch durch Deaktivierung der `pdfa`-Option genutzt werden.

6.3 Geschachtelte Akzente (Inkompatibilität zwischen `pdfx` und `amsmath`)

Es ist nach aktuellem Paketstatus (v3.04) nicht möglich beide Pakete in Kombination ohne weitere Anpassungen so zu verwenden, dass eine Schachtelung von Akzenten möglich ist. Das folgende Beispiel wirft Fehler, sobald `amsmath` direkt oder indirekt geladen wird.

```
\dot{\hat{x}}\$
```

Die zugehörige Diskussion findet sich im `tuda-ci` Repository (https://github.com/tudace/tuda_latex_templates/issues/78). Da sich diese Problematik nicht direkt aus `tuda-ci` sondern aus der Konstruktion der beiden Pakete untereinander ergibt, ist es leider nicht möglich pauschal eine Anpassung in `tuda-ci` zu implementieren. Unter dem angegebenen Link findet sich jedoch ein möglicher Workaround, sobald beide Varianten benötigt werden.

6.4 Möglicher Option Clash mit `microtype`

Das Paket `microtype` wird im Falle der Nutzung von pdf_{La}TeX oder einer erzwungenen Nutzung von Type1 Schriftarten automatisch geladen, da in der verwendeten Schriftart die Ligaturen für Kapitalchen deaktiviert werden müssen, um Probleme zu vermeiden (vgl. https://github.com/tudace/tuda_latex_templates/issues/144).

Falls es zwingend notwendig ist die Type 1 Schriftart zu verwenden, ist es möglich weitere Optionen vor dem Laden der Dokumentenklasse an `microtype` zu übergeben.

```
\PassOptionsToPackage{<microtype Optionen>}{microtype}
\documentclass{...}
```

Literatur

- [1] *Das Bild der TU Darmstadt. Corporate Design Handbuch*. Juni 2011. URL: https://www.intern.tu-darmstadt.de/media/medien_stabsstelle_km/services/medien_cd/das_bild_der_tu_darmstadt.pdf (besucht am 01.05.2020).
- [2] *IMRAD*. Wikipedia. URL: <https://en.wikipedia.org/wiki/IMRAD> (besucht am 20.12.2018).
- [3] Markus Kohm. *KOMA-Script. Die Anleitung*. 14. Jan. 2019. URL: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/doc/scrguide.pdf> (besucht am 05.02.2019).