

4.6 Schreiben mit dem Computer über den Computer. Artikel in der Informatik

Dietrich Paulus

In der Gesellschaft ist der Begriff „Informatik“ untrennbar mit Computern verbunden. Diese enge Verknüpfung ist im Prinzip zu einschränkend, denn Informatik als die Lehre von der Informationsverarbeitung bedeutet nicht ausschließlich die Beschäftigung mit der technischen Realisierung von Rechenanlagen. Dennoch hat auch die Informatik heute ein Selbstverständnis entwickelt, das den gesamten Bogen von Informationsaufnahme über Sensoren, Informationsverarbeitung durch Prozessoren bis hin zur Interpretation von Daten gerade durch Computer umfasst und diese Disziplin technisch und mathematisch beschreibt. Die meisten Veröffentlichungen in der Informatik sind also nicht nur mit dem Computer erstellt, sondern beschäftigen sich zudem mit dem Gebrauch von Computern. Damit ist der Rechner Gegenstand der Publikation *und* der Forschung. Grundlegende Tätigkeiten in der Informatik sind die Spezifikation, Dokumentation und Realisierung von Computerprogrammen. Diese Techniken der Softwareentwicklung lassen sich auch auf die Entwicklung eines Textdokuments, das zur Veröffentlichung vorgesehen ist, übertragen. Da wissenschaftliche Publikationen in der Informatik vornehmlich von InformatikerInnen gelesen werden, werden stärker noch als in anderen Disziplinen Inhalt *und* Form des Schriftstücks zur Beurteilung herangezogen. Die Beherrschung aller modernen Techniken ist damit entscheidend für den Erfolg der Veröffentlichung.

Standard-Aufbau

In der Softwareentwicklung wird ein Problem zunächst präzise spezifiziert; die Realisierung in Form eines Programms wird geplant und dann umgesetzt. Ebenso wird im Idealfall eine Veröffentlichung zunächst geplant, gegliedert und dann von „vorne nach hinten“ formuliert. In der Realität sind – ebenso wie in der Softwareentwicklung – meist viele Durchgänge erforderlich, bis alle Details stimmen, und damit erfährt der Artikel zahlreiche Überarbeitungen. Die Gliederung bleibt dabei in der Regel unverändert, und die logische Abfolge der Argumente und Ergebnisse muss nicht reorganisiert werden. Gute Veröffentlichungen benötigen

Zeit für die Entstehung. Mit jedem Durchgang wächst die Qualität des Textes (vgl. in vorliegendem Band *Wasch mich, aber mach mich nicht nass. Kürzen und Korrigieren* und den Entwicklungszyklus in *When in Rome do as the Romans do. Successfully Publishing in English*).

Beiträge in der Informatik zu Tagungen oder Zeitschriften sind in der Gliederung meist hochgradig normiert, wie der nachfolgenden Tabelle zu entnehmen ist. Die Prozentangaben sind als Richtwert zu verstehen und in dieser Form an keiner Stelle vorgeschrieben.

Tabelle 1 Standardaufteilung eines Artikels in der Informatik

	praktische Arbeiten	theoretische Arbeiten
Kurzfassung	kurz (150–200 Wörter)	kurz (150–200 Wörter)
1. Einleitung	15%	20%
2. Stand der Technik	20%	20%
3. eigener Ansatz	40%	50%
4. Experimente	20%	0%
5. Zusammenfassung	5%	10%
Literaturangaben	selten mehr als eine Seite	selten mehr als eine Seite

Quelle: eigene Darstellung

Durch diese Standardisierungen wird den GutachterInnen der faire Vergleich von Beiträgen erleichtert. Für LeserInnen ist wiederum sichergestellt, dass die Beiträge den Normen der Gemeinschaft entsprechen, beispielsweise dass der Stand der Technik ausreichend gewürdigt, d. h. beschrieben, wurde.

Von geringen Variationen abgesehen, hat sich in nahezu allen Veröffentlichungen der Informatik dieser Standardaufbau durchgesetzt, der selbst dann, wenn er nicht explizit gefordert wird, doch von LeserInnen und GutachterInnen erwartet wird. Es ist bekannt, dass LeserInnen beim Stöbern nach Literatur meist zunächst die Kurzfassung und anschließend die Zusammenfassung einer Arbeit lesen. Nur wenn diese beiden Abschnitte interessant erscheinen, werden auch die anderen betrachtet. Brillante SchreiberInnen und Beiträge von überragender Bedeutung (beispielsweise Canny 1986) können im Einzelfall von diesen Normen abweichen, aber für einen „normalen“ Beitrag empfiehlt es sich, genau dieser Struktur analog zur Tabelle zu folgen. In den Überschriften zu Abschnitt 2 (*Stand der Technik*) und 3 (*eigener Ansatz*) können die Themen der Untersuchung spezifiziert und individualisiert wiedergegeben werden, indem beispielsweise in einer Arbeit über Compilerbau die Überschriften „Traditionelle Ansätze“ und „Ein neues System zur schnellen Syntaxanalyse“ gewählt werden. Die anderen Überschriften ent-

sprechen meistens den standardisierten Vorgaben in der Tabelle. Ein Beispiel für eine praktische Arbeit nach diesem Aufbau liefern Bin Luo und Edwin Hancock (2001) oder Yuri Gurevich (1990) für eine theoretische Arbeit.

Im Folgenden wird dargelegt, wie die einzelnen Elemente eines Fachartikels aufgebaut sein sollten:

Die *Kurzfassung* soll knapp gehalten sein und in 150 bis 200 Wörtern den Inhalt des Beitrags darlegen. Sie soll keine Referenzen und Zitate enthalten, damit sie auch verständlich ist, ohne dass die LeserInnen das gesamte Schriftstück vorliegen haben. Dies ermöglicht es den HerausgeberInnen, die Kurzfassungen (Abstracts) von Zeitschriftenbeiträgen, Aufsätzen in Büchern und Tagungsbänden (*conference proceedings*) auf Suchmaschinen einzustellen und den Artikel davon getrennt zum Verkauf anzubieten.¹ Nicht zuletzt aus diesem Grund muss dieser Gliederungspunkt Ergebnisse und Inhalte zusammenfassen, vermitteln, *wie* die Ergebnisse erzielt wurden und möglichst konkret die Erkenntnisse des Beitrags in Kurzform liefern.

Die *Einleitung* führt die LeserInnen in das Thema des Beitrags ein, stellt den Kontext her und erläutert den Nutzen der Ergebnisse oder Untersuchungen. Am Ende der Einleitung soll eine Erläuterung zum Aufbau des Artikels folgen. Dies mag redundant erscheinen, wenn die normierte Gliederung gewählt wird – es hat sich aber so eingebürgert.

Im Abschnitt *Stand der Technik* belegen die AutorInnen, dass ihre eigenen Arbeiten nicht in ähnlicher Form schon von anderen veröffentlicht wurden. Alle relevanten Vorarbeiten anderer AutorInnen zu Themen der Publikation sollten erwähnt werden. Dieser Abschnitt kann auch nach hinten verschoben werden, um den Bezug zur eigenen Arbeit herzustellen.

In *eigener Ansatz* wird erklärt, wie sich die eigenen Arbeiten von denen anderer AutorInnen unterscheiden. Die eigenen Ideen und Verfahren werden zunächst präzise dargelegt. Dazu werden bei Bedarf Formeln, Beweise, Diagramme, Programmfragmente, Algorithmen in Pseudocode und natürlich eine präzise Sprache verwendet. Literaturangaben zu eigenen Vorarbeiten werden hier aufgeführt. Anschließend wird der eigene Ansatz mit anderen Vorschlägen verglichen, die wiederum mit Literaturangaben belegt werden.

Wenn es sich um eine praktische Arbeit handelt, die auf empirisch ermittelten Daten beruht, wird im Abschnitt *Experimente* dokumentiert, welche Daten verwendet, welche Ergebnisse erzielt wurden und wie sich diese von den Ergebnissen unterscheiden, die anderenorts veröffentlicht wurden oder die mit Verfahren anderer AutorInnen erreicht werden können. Insbesondere die Vergleichbarkeit und

¹ Die AutorInnen stimmen in der Regel dieser Praxis schriftlich zu, sobald sie ein Manuskript an einen Verlag übermitteln.

Reproduzierbarkeit der Experimente sollte einen sehr hohen Stellenwert haben. Je nach Thema der Arbeit können an dieser Stelle auch Evaluationen oder eine Falluntersuchung (*case study*) stehen.

In der *Zusammenfassung* werden die wesentlichen Ergebnisse noch einmal kompakt erläutert, und mögliche Erweiterungen oder zukünftige Arbeiten werden diskutiert.

Die *Literaturangaben* sollten ausgewogen die eigenen Vorarbeiten und die fremder AutorInnen dokumentieren, wobei die Anzahl der Fremdzitate in der Regel deutlich oberhalb der Hälfte liegen sollte. Wer zu viele eigene Zitate angibt, wirkt unter Umständen borniert.

Der Veröffentlichungsprozess

In der Informatik ist es üblich, eine erste Version einer Veröffentlichung auf einem Workshop zu publizieren. Eine erweiterte und überarbeitete Version wird danach auf einer Tagung vorgestellt. Schließlich liefern mehrere Veröffentlichungen das Material für einen Zeitschriftenbeitrag, sofern die Zeitschrift nicht unveröffentlichtes Material einfordert. In manchen Fällen wird eine Idee auch schnell als „Technischer Report“² publiziert, bevor sie über mehrere Stufen in immer höherrangigen Organen veröffentlicht wird. Dies unterstreicht, dass ein Text in verschiedenen Versionen mehrfach verwendet werden kann. Ein Beispiel liefert Carlo Tomasi mit Technischen Reports (Tomasi/Kanade 1991a; Tomasi/Kanade 1991b) und der Veröffentlichung in einer namhaften Konferenz (Shi/Tomasi 1994). Auch Canny (1986) wurde zuerst als Technischer Report (Canny 1983) veröffentlicht.

Im Gegensatz zu anderen Disziplinen ist die Reihenfolge der Namen in der Autorenangabe nicht normiert. Der Hauptautor/Die Hauptautorin steht in der Regel vorne, die KoautorInnen werden in unterschiedlicher Form sortiert, beispielsweise alphabetisch oder geordnet nach dem Umfang des Beitrags zur Veröffentlichung. Während andere Disziplinen auch eine explizite Rangfolge bei der Bedeutung von Konferenzen und Zeitschriften definieren (über den sogenannten *impact factor*), so ist auch hier in der Informatik eher ein informelles Verständnis der Qualität einer Veröffentlichung zu beobachten. Jede der Teildisziplinen in der Informatik hat ihre eigenen Fachtagungen und Fachzeitschriften. Abgesehen davon, dass die IEEE³ und ACM als Qualitätssiegel gelten, sind wenig generelle Regeln festzu-

² Ein Technischer Report wird von der Forschungseinrichtung herausgegeben und in der Regel mit laufenden Nummern versehen. Diese Veröffentlichungen sind dann nicht über den Buchhandel, sondern nur über die Forschungseinrichtung (oder über das Internet) zu beziehen.

³ Viele wichtige Ergebnisse der Informatik werden in Organen der IEEE (Institute of Electrical and Electronic Engineers) und ACM (Association for Computing Machinery) veröffentlicht.

stellen. Beispielsweise sind Tagungen der SPIE (siehe hierzu <http://www.spie.org>) in manchen Teildisziplinen hoch angesehen, in anderen Teildisziplinen dagegen eher als Tagung mit geringer Bedeutung verstanden.

Auch der *review*-Prozess ist nur bedingt aussagekräftig. In der Regel sind Tagungen, die einen Vollbeitrag zur Begutachtung einfordern, besser angesehen. Es gibt aber auch hochrangige Konferenzen, zu denen Beiträge auf Basis einer erweiterten Kurzfassung angenommen werden. Dem Neuling im Fachgebiet sei es empfohlen, die Einschätzung von erfahrenen KollegInnen vor der Auswahl einer Konferenz oder einer Zeitschrift einzuholen.

Ablaufplan

Nachfolgend wird ein Ablaufplan für das Vorgehen bei der Publikation eines Fachartikels in der Informatik dargelegt. In den meisten Fällen werden mehrere AutorInnen an einem Artikel beteiligt sein. Daher ist eine strukturierte Dokumentenverwaltung in einem so genannten *repository* essenziell.

- 1) Publikationsorgan/Ziel auswählen
- 2) Gliederung und Inhalt festlegen, evtl. KoautorInnen verpflichten, *repository* anlegen
- 3) Dokumentenstruktur definieren
 - 3.1) ggf. vorhandenes Material einpflegen
- 4) Dokumenterstellung
 - 4.1) Dokument ins *repository* einfügen
 - 4.2) überarbeiten
 - 4.3) forschen/experimentieren/mit KoautorInnen diskutieren
- 5) wenn noch nicht zufrieden und noch Zeit vorhanden, zurück zu 4)
- 6) Dokument ins *repository* legen und markieren (*tag*)
- 7) einreichen zur Begutachtung
- 8) bei Annahme
 - 8.1) ggf. Hinweise der GutachterInnen einarbeiten
 - 8.2) Dokument ins *repository* legen und markieren
 - 8.3) finale Version einreichen
- 9) bei Ablehnung
 - 9.1) Hinweise der GutachterInnen einarbeiten
 - 9.2) Dokument ins *repository* und markieren
 - 9.3) Wiederverwendung in kommender Veröffentlichung (siehe 3.1)

Seltener kommt es vor, dass am Anfang des Prozesses die wissenschaftliche Idee steht und diese zu Papier gebracht wird, ohne dass eine konkrete Veröffent-

lichung anvisiert wird. Diese Vorgehensweise ähnelt einer *bottom-up*-Entwicklung, während der oben genannte Prozess einer *top-down*-Vorgehensweise entspricht. Sobald nun zu einem späteren Zeitpunkt eine Veröffentlichung geplant ist, müssen die Dokumente angepasst werden, um in Format, Umfang und der thematischen Ausrichtung den neuen Vorgaben zu entsprechen.

Der Entstehungsprozess

Publizieren in der Informatik hat viel mit der Durchführung eines Softwareprojekts und mit Programmieren zu tun. In den meisten Fällen handelt es sich um einen iterativen Prozess, wie er im Ablaufplan des vorherigen Abschnitts skizziert ist. Bei der Dokumentation der Ergebnisse entstehen beispielsweise neue Erkenntnisse, die eine Erweiterung des Stands der Technik erfordern. Da InformatikerInnen auch das Programmieren als Handwerk beherrschen, wird natürlicherweise eine Publikation ebenfalls das Ergebnis eines „Programms“. Dieser Ansatz wird technisch durch die Werkzeuge zur Veröffentlichung unterstützt – vorausgesetzt es werden die angemessenen Werkzeuge verwendet (vgl. in vorliegendem Band *Wie und womit? Programme zur Erstellung und Verarbeitung von Texten*). Alle Teile des Textes, die automatisch entstehen können, sollten in der Regel auch mit Werkzeugunterstützung automatisch generiert werden. Dies gilt für Nummerierung von Listen, Gliederung, Abbildungsnummern und Verzeichnisse. Verweise auf Abbildungen und Abschnitte, Literaturzitate und Gleichungsnummern werden nur dann vollständig und konsistent, wenn sie vom Textverarbeitungswerkzeug erstellt und überprüft werden. Das Literaturverzeichnis wird aus einer Datenbank erzeugt – nur so ist sichergestellt, dass alle zitierten Arbeiten aufgeführt und auch keine Arbeiten angegeben werden, die im Text keine Erwähnung finden.

Da die Entstehung des Dokuments in mehreren Schleifendurchläufen geschieht, wird durch diese Techniken immer gewährleistet, dass die Verweise korrekt sind. Wird das Dokument in einen größeren Beitrag, ein Buch oder eine Zeitschrift eingefügt, so passen sich die Verweise automatisch an: Seitenangaben, Verweise und auch des Format und der Stil der Referenzen werden vom Globaldokument bestimmt.

Wie in der Programmierung werden auch für die symbolischen Verweise (auf Abbildungen, Gleichungen etc.) *style guides* verfasst, die – wie im Fall von Variablennamen in der Programmierung – die Eindeutigkeit der Variablennamen weitgehend sicherstellen.

Werkzeuge

Mit der Verbreitung der modernen Satztechniken steigt der Anspruch an die technische Qualität der Veröffentlichungen. Perfekte Diagramme und Formelsatz sind heute Grundvoraussetzung für eine erfolgreiche Publikation. Dies bedeutet, dass ungeeignete Werkzeuge für den Textsatz ausgeschlossen werden müssen. Ist ein Werkzeug für die Erstellung von Briefen im Bürobereich konzipiert, so ist es prinzipiell ungeeignet zur Produktion von Büchern, insbesondere wenn die Kapitel von unterschiedlichen AutorInnen zeitgleich erstellt werden sollen. Der Blick auf eine solche Publikation als ein Softwareprojekt weist den richtigen Weg zu den richtigen Werkzeugen: Verteilte Versionskontrollsysteme, ein definierter Prozess zur Übergabe von Änderungen an die anderen AutorInnen, eine automatische Dokumentation der Entstehungshistorie, die Möglichkeit zur Aufhebung von Änderungen, die Visualisierungsmöglichkeit für Textdifferenzen etc. werden von jedem Programmierer/jeder Programmiererin gefordert und genutzt. Diese Technik sollte auch zur Erstellung zumindest von umfangreichen Dokumenten verwendet werden, selbst dann, wenn nur ein Autor/eine Autorin beteiligt ist.

Werkzeuge, die einen solchen Entwicklungsvorgang ermöglichen, sind in der Regel auch professionell, was das Layout angeht. Formeln müssen diverse Sonderzeichen in allen Größen ermöglichen – selbstverständlich in einheitlichem Font.

Mathematik und Formeln

Eine Veröffentlichung, die sich um Mathematik dreht, muss ein Satzsystem verwenden, das speziell für den Formelsatz geeignet ist. Auch hier ist die Sicht auf das Publizieren als ein Programmiervorgang nützlich. Im *Stand der Technik* werden oft Formeln verwendet, die aus fremden Veröffentlichungen stammen. Werden nun mehrere Formeln verschiedener AutorInnen verglichen oder aus unterschiedlichen Bereichen im eigenen Ansatz zusammengeführt, so kommt es unweigerlich zu Konflikten in der Verwendung von Symbolen in den Formeln. Durch die Definition von symbolischen Makronamen für die gedruckten Symbole lassen sich diese Konflikte an nur einer Stelle im Dokument zusammenfassen und einfach lösen. Dies wird vor allem dann notwendig, wenn umfangreiche Dokumente wie eine Dissertation erstellt werden. Generell muss die Notation innerhalb eines Dokuments konsistent sein und Symbole sollten einheitlich im Text, in Diagrammen und Formeln eingesetzt werden, auch im Hinblick auf den verwendeten Font. Wie in einem Computerprogramm müssen im Gesamtdokument die Bezeichner einheitlich verwendet werden.

Diagramme und Bilder

Diagramme müssen mit großer Sorgfalt erstellt werden, denn auch hier ist perfektes Layout erforderlich. Hierbei liefern automatische Werkzeuge jedoch nicht immer die besten Ergebnisse. Oft ist eine programmierte Grafik – beispielsweise mit PGF – der Ausgabe eines interaktiven Grafik-Werkzeugs überlegen. Bei der Erstellung von Grafiken sind Grundregeln gültig, deren Übertretung eine Veröffentlichung oft unmöglich macht. In der Informatik werden häufig Diagramme eingesetzt, die Kästchen und Pfeile enthalten. Grundsätzlich sollten diese Diagramme einheitlich formatiert sein. Pfeile beginnen exakt am Kästchen und enden exakt dort, Pfeilspitzen sind einheitlich gestaltet, Texte sind optimal in die Kästchen eingepasst.

Solche Diagramme entstehen oft mit Werkzeugen, die nicht auf das Textverarbeitungswerkzeug abgestimmt sind. Da im Prozess der Publikation das Dokument in der Größe zum Druck verändert werden kann, ist es unerlässlich, dass auch die eingebundenen Diagramme in der Größe veränderbar sind. Für Diagramme sind Vektorgrafiken obligatorisch. Der Export einer Grafik als Bild (*bitmap*) und die Verwendung eines solchen Bilds im Text führen in vielen Fällen zu minderwertiger Druckwiedergabe und sind daher ausgeschlossen. Kompressionsartefakte in Bildern, die aus Grafiken erzeugt wurden, zählen schon fast zu Peinlichkeiten. Einige Verlage (beispielsweise IEEE-Zeitschriften) verweigern gar die Annahme von Dokumenten, in denen Grafiken als Bilder beigefügt sind. Einige der genannten Fehler sind in der folgenden Abbildung demonstriert.

Abbildung 1 Typische Fehler in Diagrammen



Quelle: eigene Darstellung

Ein nicht zu vernachlässigender Aufwand ist erforderlich, um Quellenangaben vollständig und korrekt in eine Veröffentlichung zu integrieren. Die Verwendung von Datenbanken bietet sich hier an, insbesondere was die Literaturangaben angeht (vgl. in vorliegendem Band *Anleitung zur Selbstmedikation. Literaturverwaltung*). Teilweise können diese Datenbanken auch aus öffentlich zugänglichen Quellen mit Einträgen versorgt werden. Beispielsweise sind sehr viele Informatikartikel zentral archiviert (vgl. dazu *The BLBP Computer Science Bibliography*, <http://dblp.uni-trier.de/>) und können als *BibTeX*-Einträge von dort geladen werden.

Internetquellen (so genannte URLs), die nicht als Onlinejournale editiert veröffentlicht wurden und die nicht über eine ISBN oder ISSN verfügen, sollten getrennt von Printmedien referenziert werden, beispielsweise als Fußnoten. Von Hinweisen auf Wikipedia ist in wissenschaftlichen Arbeiten generell abzusehen.

Während Fußnoten in den Geisteswissenschaften auch für Literaturangaben genutzt werden, wird in der technischen Literatur durchgängig die Referenz direkt im Satz eingefügt und kann grammatikalisch auch als Subjekt oder Objekt verwendet werden („Canny 1986 stellt eine herausragende Arbeit dar“).

Sprache

Technische Dokumente werden in einfacher Sprache verfasst. Lange komplexe Sätze sind ebenso zu vermeiden wie ein lyrischer Stil. Dies trifft insbesondere auf Veröffentlichungen in englischer Sprache zu. Hier gilt die Regel, so zu schreiben, wie man sprechen würde (vgl. Alred/Brusaw/Oliou 2006). Während sich im Deutschen die Verwendung von Passivkonstruktionen eingebürgert hat, werden im Englischen Sätze in Ich-Form und im Aktiv bevorzugt (vgl. Lindsell-Roberts 2001). Weitere Hinweise für englischsprachige technische Veröffentlichungen finden sich in Alred/Brusaw/Oliu (2006).

Dieser prinzipielle Unterschied ist vor allem dann von Bedeutung, wenn ein deutsches Dokument „wiederverwendet“ und für eine nachfolgende Veröffentlichung ins Englische übersetzt wird (Zum Schreiben und Publizieren auf Englisch vgl. in vorliegendem Band *When in Rome do as the Romans do. Successfully publishing in English*).

Wiederverwertung

Abhängig davon, welche Anforderungen durch das Organ formuliert werden, in dem veröffentlicht werden soll, können unterschiedliche Teile früherer Arbeiten wiederverwertet werden. Dies entspricht vollständig der Sichtweise einer Publikation als einem Softwaremodul, in dem auch eine hohe Wiederverwertbarkeit angestrebt wird. Auf jeden Fall lassen sich Diagramme und Bilder in Vorträgen erneut präsentieren. Geeignete Werkzeuge – wie *LaTeX* – erlauben die Wiederverwendung von Textteilen oder Diagrammen in einer Technik, die vergleichbar ist zu einer modularen Programmentwicklung.

Während in der Vergangenheit gefordert wurde, dass eine Dissertation bislang unveröffentlichte Erkenntnisse enthielt, hat sich heute die Situation ins Gegenteil gewandelt: Die Deutsche Forschungsgemeinschaft (DFG) als hochrangiges und einflussreiches Organ fordert, dass Ergebnisse laufend publiziert werden:

„Die DFG erwartet, dass die mit ihren Mitteln finanzierten Forschungsergebnisse publiziert und dabei möglichst auch digital veröffentlicht und für den entgeltfreien Zugriff im Internet (Open Access) verfügbar gemacht werden“ (Deutsche Forschungsgemeinschaft 2009).

In den meisten Promotionsordnungen im Bereich Informatik sind heute keine Klauseln mehr zu finden, die eine Vorveröffentlichung verbieten. Eigene Vorarbeiten werden in die Dissertation eingefügt und erhöhen das Gewicht der Arbeit.

Zusammenfassung

Das Verständnis des Textentstehungsprozesses als eine Tätigkeit, die der Durchführung eines Projekts (im Sinne der Softwaretechnik) ähnlich ist, ist der Entstehung eines Dokuments in der Informatik angemessen. Die Entwicklungswerkzeuge zur Programmierung lassen sich in ähnlicher Form für die Programmierung von Dokumenten und Texten verwenden, die hohen technischen Anforderungen genügen. Die Qualität der Inhalte kann das selbstverständlich nicht ersetzen. Doch ohne gute Darstellungsform ist heute guter Inhalt nicht mehr zur Publikation geeignet. Auf dem Weg zur Dissertation ist es heute unerlässlich, frühzeitig zu publizieren und Erfahrung mit wissenschaftlichem Schreiben zu sammeln.

Literatur

- Alred, Gerald J./Brusaw, Charles T./Oliu, Walter E. (2006): *The Handbook of Technical Writing*. Eighth Edition. Boston, MA: Bedford/St. Martin's
- Canny, John F. (1983): *Finding edges and lines in images*. Cambridge, Mass.: M. I. T. Artificial Intell. Lab.
- Canny, John F. (1986): *A computational approach to edge detection*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8. 6: 679–698
- Deutsche Forschungsgemeinschaft (2009): *Verwendungsrichtlinien. Sachbeihilfen mit Leitfaden für Abschlussberichte und Regeln guter wissenschaftlicher Praxis* (http://www.dfg.de/forschungsfoerderung/formulare/download/2_01.pdf; 30.11.2009)
- Gurevich, Yuri (1990): *Algebraic Operational Semantics and Occam*. In: *CSL'89, 3rd Workshop on Computer Science Logic, Lecture Notes in Computer Science*. Berlin: Springer: 176–191
- Lindsell-Roberts, Sheryll (2001): *Technical Writing for Dummies*. New York: Hungry Minds
- Luo, Bin/Hancock, Edwin R. (2001): *Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition*. In: *Transactions on Pattern Analysis and Machine Intelligence* 23. 10: 1120–1136

- Tomasi, Carlo/Kanade, Takeo (1991a): Shape and Motion from Image Streams: a Factorization Method – part 2: Point Features in 3D Motion. Techreport, Carnegie, Mellon University, CMU (<http://citeseer.ist.psu.edu/tomasi91shape.html>; 30.11.2009)
- Tomasi, Carlo/Kanade, Takeo (1991b): Detection and Tracking of Point Features. Techreport, Carnegie Mellon University, CMU (<http://www.ces.clemson.edu/~stb/kl/tomasi-kanade-techreport-1991.pdf>; 30.11.2009)
- Shi, Jianbo/Tomasi, Carlo (1994): Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), Seattle: 593–600 (<http://www.cs.duke.edu/~tomasi/papers/shi/shiCvpr94.pdf>; 30.11.2009)