

Richtlinien für Bakkalaureats- und Master-Arbeiten am CGV

Sven Havemann & andere
Institut für ComputerGraphik und WissensVisualisierung (CGV), TU Graz
www.cgvtugraz.at

11. Januar 2010

Zusammenfassung

Wir geben hier einige Richtlinien für praktische und schriftliche Abschlussarbeiten an unserem Institut. Die Richtlinien sind nicht absolut bindend, aber man kann nachher nicht sagen, man hätte von nichts gewusst! Massgebend ist im Einzelfall die unfehlbare Objektivität des Betreuers. Damit der Betreuer aber nicht alles immer wieder erzählen muss, haben wir hier einmal einige wesentliche Punkte gesammelt und im Zusammenhang aufgeschrieben.

1 Die praktische Arbeit

- Der Betreuer formuliert eine Problemstellung bzw. ein Ziel, teilweise bewusst vage, und übernimmt die Rolle des “Kunden”, der eine Forschungs-Dienstleistung vom Studierenden haben möchte
- Herausforderung: Studierender muss Problem in wohldefinierte Aufgabenstellung transformieren
 - Bakk-Arbeit: Betreuer stellt sicher, daß der Studierende alle wesentlichen Mittel zur Lösung hat, im Prinzip sollte klar sein was zu tun ist
 - Master-Arbeit: Mittel zur Lösung der Aufgabe müssen zu grossem Teil selber erarbeitet werden; wesentlich höherer Kreativ-Anteil als beim Bakk
- Beispielrechnung Zeitplan – Genaues bitte den Studienrichtlinien entnehmen, denn es gibt da eine große Schwankungsbreite (1 ECTS = 25-30 h)
 - Bakk Telematik:
6 ECTS-Punkte, ist 150-180 h Arbeit, entspricht ca. 4-5 Wochen Vollzeit (bei 40 h / Woche)
 - Bakk Software-Entwicklung Wirtschaft:
15 ECTS-Punkte, ist 375-450 h Arbeit, entspricht ca. 11-12 Wochen Vollzeit (bei 40 h / Woche)
- Wir haben eine Empfehlung für den ungefähren Zeitbedarf, auch wenn es im Einzelfall zumeist davon abweicht. Bindend dagegen ist der Zeitplan, den man mit dem Betreuer ausmacht. Seine Einhaltung ist auch Gegenstand der Bewertung, denn Lernziel ist ja auch, die eigene Leistungsfähigkeit realistisch einschätzen zu können.
 - Richtwerte wären für den Zeitplan wenn die Bakk-Arbeit 3 Monate dauert:
 - 4 Wochen Einarbeitung+Recherche
 - 2 Wochen Erarbeiten eines Arbeitsplans
 - 6 Wochen Implementation
 - 2 Wochen für Testen, Zweifelsfälle, Sourcen polieren
 - Richtwerte für den Zeitplan bei Master von 6 Monaten:
 - 6 Wochen Einarbeitung+Recherche,
 - 2 Wochen Erarbeiten eines Arbeitsplans,
 - 12 Wochen Implementation,
 - 4 Wochen für Testen, Zweifelsfälle, Sourcen polieren
- Schon bei der Recherche alle gelesenen Quellen kurz dokumentieren, jeweils mindestens ein Absatz (für Kapitel 2)
- Wichtig: Schon bei der praktischen Arbeit das spätere Aufschreiben vorbereiten, im Nachhinein läßt sich Vieles nicht mehr ohne weiteres rekonstruieren:
 - Designentscheidungen dokumentieren
 - Tests und Zwischenergebnisse dokumentieren

- Fehlschläge dokumentieren, ebenso wie sich daraus ergebende Konsequenzen
- Viele aussagekräftige Fotos bzw. Screenshots machen

2 Die schriftliche Arbeit

Zweck der Arbeit ist, den Nachweis zu führen, daß der Kandidat ingenieurmäßig, selbständig und wissenschaftlich arbeiten kann.

- **ingenieurmäßig** heisst, daß mit Augenmaß das Machbare angestrebt und Schwieriges in Einfacheres zerlegt wird
- **selbständig** heisst, daß wesentliche Design-Entscheidungen eigenständig getroffen werden
- **wissenschaftlich** heisst, daß sämtliche Design-Entscheidungen nachvollziehbar begründet sind (nicht “aus dem Bauch heraus”), unter Berücksichtigung des aktuellen *state of the art*

Bei Bewertung steht in jedem Fall die schriftliche Arbeit im Vordergrund. Codequalität als solche ist nicht Gegenstand der Benotung. Code wird aber auch indirekt bewertet, weil er Teil des Inhalts der schriftlichen Arbeit ist (API-Beschreibung, Code-Beispiele).

Im Grundatz gilt auf jeden Fall für die schriftliche Arbeit: Man soll genau das Dokument verfertigen, das man selber gerne gehabt hätte, als man mit der Arbeit anfang. Diese Leitlinie beantwortet auch viele Fragen wie: Ist das ausführlich genug? Soll ich auf dieses oder jenes noch eingehen? Soll ich auch schildern, was danebengegangen ist? Wieviele Beispiele soll ich bringen? – Man muss dazu nur versuchen, die eigene Arbeit mit fremden Augen zu lesen.

Zum generellen Aufbau: Eine wissenschaftliche Arbeit sollte die spannende Geschichte einer Problemlösung erzählen. Dies soll sich auch im Aufbau der Arbeit widerspiegeln. Sprich, um ein konkretes Beispiel zu geben: Zu Beginn wird ein wichtiges aber quasi unlösbares Problem beschrieben (Motivation, Kapitel 1), das dann tapfer in lösbare Teilprobleme zerlegt wird, wo man auch schaut was einem schon weiterhelfen könnte (Kapitel 2). Dann kommt die zündende Idee, aus der sich ein Plan entwickelt (Kapitel 3), der dann konsequent verfolgt und umgesetzt wird (Kapitel 4). Schliesslich kommt das Happy-End, wo man nachweist bis zu welchem Grad das Problem gelöst ist; und daraus ergibt sich auch die Problemstellung für die Fortsetzung in einem hypothetischen nächsten Teil (Kapitel 5). Der Aufbau ist im Prinzip völlig gleich für Bakk- und Master-Arbeiten, nur der Umfang unterscheidet sich (erfahrungsgemäß 25 vs. >50 Seiten, Details sind der Studienordnung zu entnehmen bzw. mit dem Betreuer auszuverhandeln).

Ein Tipp: In jedem Abschnitt bevor man mit dem Schreiben beginnt erst die Bilder machen, die das zeigen, was man später textuell beschreibt. Beim Machen der Bilder ergibt sich die Geschichte dazu schon zum Teil von selbst. Wenn man möchte, daß die Arbeit bleibenden Wert hat und auch international gelesen werden kann, schreibt man zweckmäßigerweise auf Englisch. Das ist eine Empfehlung beim Bakk und eine deutliche Empfehlung beim Master.

Zum Sprachlichen: Die Worte werden bei der Beurteilung auf die Goldwaage gelegt! Daher bemühe man sich, den Text frei zu halten von Rechtschreibfehlern, stilistischen Verfehlungen, mißverständlichen Aussagen. Präzise, klar, unmissverständlich ist gefragt; aber auch nichts wegzulassen, das zum Verständnis wichtig ist. Es sollten objektive Fakten (*es ist, es wurde*) im Vordergrund stehen, nicht Subjektives oder Personen (*ich habe*).

Bilder sollten generell weissen Hintergrund haben. Dunkle Farben oder gar schwarz nur verwenden wenn wirklich nötig. Eine LaTeX-Gerüst für solche Abgaben findet sich unter [LaTeX-Vorlage-Abschlussarbeit-2007-01-12.zip](#). Dort finden sich auch ergänzende Hinweise.

Überblick

- Wie “Klappentext” bei einem Roman: Zu Beginn, halbe bis eine Seite
- Elterntaugliche Beschreibung: In normalen, klar verständlichen Worten Problemstellung und Lösung schildern.
- Aufgabenstellung, Einordnung in einen Bereich

Kapitel 1: Introduction

- Herausforderung, generelle Problembeschreibung, Motivation
- Präzisierung der Problembeschreibung
- Alle Fachbegriffen müssen schön eingeführt und erklärt werden, beim ersten Auftauchen kursiv
- Aufbau des restlichen Dokuments kurz beschreiben, falls nicht eindeutig (bzw. innerhalb des Kapitels)

Kapitel 2: Related Work

Schwierigkeit bei Related Work ist ein Henne-Ei-Problem: Erst muss man die Aufteilung in Teilprobleme vornehmen, um dann zu den Teilproblemen existierende Lösungen zu suchen. Zum Teil legen die existierenden Lösungen aber auch nahe, wie die Aufteilung in Teilprobleme vorzunehmen ist.

- Analyse und Realisierung der Teilprobleme: Was brauchen wir alles?
- Sämtliche relevante Literatur finden und den Zusammenhang zur Problemstellung besprechen: Was kann man nutzen, was kann man nicht nutzen, und warum
- Damit Designentscheidungen begründen

Kapitel 3: Technisch (z.B. Entwurf, Algorithmus)

Die Lösungsidee auf den Punkt bringen, und zwar in eher theoretischer bzw. konzeptioneller Form. Man soll nachweisen, dass man in der Lage ist, "geradeaus" zu denken. Typische Ansätze wären z.B.:

- Was könnte man im Prinzip überhaupt tun, und was würde das kosten (Aufwandsabschätzung)
- Wie könnte man z.B. eine Zielfunktion formulieren, um daraus ein Optimierungsproblem zu gewinnen?
- Was ist die ground-truth (unerreichbare Ideallösung), und wie kann man dieses Ideal annähern?
- Was verliert man bei der Annäherung, wie gut kann man theoretisch überhaupt werden?
- Allteimein: Welche Trade-offs gibt es?

Kapitel 4: Technisch (z.B. Implementation)

Ingenieurmäßige Umsetzung der Lösungsidee. Titel hängt stark von der Aufgabenstellung ab.

- Zum Schluss des Kapitels API-Beschreibung: Welche Module gibt es, was sind die wesentlichen Klassen?
- Beispiele: Wie wird der Code aufgerufen, wie sieht die typische Benutzung aus?

Kapitel 5: Ergebnisse, Diskussion

Dies ist das Qualitätssicherungs-Kapitel. Es unterscheidet sich in sofern von den anderen Kapiteln, als daß man hier die Rolle wechselt: Von jemandem, der eine Arbeit durchführt, wird man zu jemandem, der eine Arbeit bewertet. Diese Bewertung sollte möglichst glaubwürdig und damit ehrlich sein. Es ist ein Zeichen von Wissenschaftlichkeit, wenn man schonungslos Grenzen und Limitationen aufdeckt, die noch bestehen.

- kritische Betrachtung, Qualitätssicherung, Ziele erreicht?
- Was hat nicht funktioniert? Wo sind die Grenzen des Ansatzes?
- Unter welchen Bedingungen funktioniert es besonders gut?
- Messbarkeit, quantitative Untersuchung: Jede Ingenieursarbeit sollte irgendwo aussagekräftige Diagramme enthalten (Zeit, Platz, Genauigkeit, Problemklassen, usw.)
- Schliesslich Fazit: Kritische Reflexion der eigenen Arbeit, bzw. des Erreichten.

3 Sonstiges

- Es ist generell ein Abschlussvortrag zu halten. Bei Master-Arbeiten wird zusätzlich ein Anfangsvortrag gehalten, wo man nach der Einarbeitungsphase die Ziele der Arbeit präsentiert – an denen man sich schließlich messen lassen muss!
- Es gibt ein internes CGV-Wiki, wo jeder Student, der eine Abschlussarbeit bearbeitet, den Fortschritt seiner Arbeit dokumentieren muss (wenn der Betreuer darauf Wert legt): <http://www.cgvtugraz.at:8080/cgvwiki>
- Wiederholbarkeit: Verfahren so präzise beschreiben, dass sie nach-implementiert werden können
- Auf Nachhaltigkeit achten
- Code-Dokumentation: Ausgewählte wesentliche Teile in der Ausarbeitung, vollständig im Header, Wichtiges in der Implementation

- Code: Fallbeispiele, die die Benutzung demonstrieren, sowohl als Testprogramme wie auch in Kapitel 4 in der Ausarbeitung
- Abgabe: Finaler Stand mit umfassendem Material in unserem AFS/archive/abgaben
 - Schriftliche Arbeit als PDF
 - PDFs von allen Artikeln, die man referenziert hat
 - Code: Sourcen, Binaries und *Readme* zur Bedienung (!)
 - Folien des Vortrages (der Vorträge)
 - Diverses (alles andere was man noch brauchen könnte)