

30 Jahre Literarische und Dokumentarische Datenverarbeitung an der Universität Tübingen - 80 Kolloquien: mehr als nur zwei Jubiläen

Ott, Wilhelm

Veröffentlichungsversion / Published Version

Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:

GESIS - Leibniz-Institut für Sozialwissenschaften

Empfohlene Zitierung / Suggested Citation:

Ott, W. (2001). 30 Jahre Literarische und Dokumentarische Datenverarbeitung an der Universität Tübingen - 80 Kolloquien: mehr als nur zwei Jubiläen. *Historical Social Research*, 26(2/3), 255-263. <https://doi.org/10.12759/hsr.26.2001.2/3.255-263>

Nutzungsbedingungen:

Dieser Text wird unter einer CC BY Lizenz (Namensnennung) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier:

<https://creativecommons.org/licenses/by/4.0/deed.de>

Terms of use:

This document is made available under a CC BY Licence (Attribution). For more information see:

<https://creativecommons.org/licenses/by/4.0>

HUMANITIES COMPUTING

30 Jahre Literarische und Dokumentarische Datenverarbeitung an der Universität Tübingen - 80 Kolloquien: mehr als nur zwei Jubiläen

*Wilhelm Ott (Tübingen)**

1. Das Umfeld

Vor ziemlich genau 40 Jahren, am 24. November 1960, fand in der Neuen Aula der Universität Tübingen mit dem "Internationalen Kolloquium über maschinelle Methoden der Literarischen Analyse und der Lexikographie" eines der weltweit ersten Kolloquien zum Einsatz der EDV in den Geisteswissenschaften statt. Es war organisiert worden vom Centro per L'Automazione dell'Analisi Linguistica e Letteraria (CAAL) in Gallarate und dessen Leiter, P. Roberto Busa. 30 Jahre später konnte er bei unserem 50. Kolloquium einen Rückblick über "Half a Century of Literary Computing: Towards a 'New' Philology" geben.

Wie schon das Kolloquium von 1960 gezeigt hat, ist an der Universität Tübingen das Klima für den Einsatz innovativer Methoden in den Geisteswissenschaften offensichtlich traditionell günstig. Dies zeigte sich sechs Jahre später wieder, als im gleichen Jahr, in dem das Rechenzentrum aus dem Mathematischen Institut ausgegliedert und eine Zentrale Einrichtung wurde, an eben diesem Rechenzentrum ein Mitarbeiter eingestellt wurde mit der Aufgabe, die Belange der geisteswissenschaftlichen Fächer wahrzunehmen. Was das hieß in einer Zeit, in der die Rechenzentren eine reine Domäne der Mathematik und Naturwissenschaften waren, als die Rechner teuer und Rechenzeit knapp war, kann man heute kaum mehr ermessen. Immerhin hat es damals am Deutschen Rechenzentrum in Darmstadt schon eine "Abteilung Nichtnumerik" gegeben; dort hatte ich selbst im Frühjahr 1966 meine ersten Gehversuche auf dem Computer unternommen.

* Aus dem Protokoll des 80. Kolloquiums über die Anwendung der Elektronischen Datenverarbeitung in den Geisteswissenschaften an der Universität Tübingen vom 18. November 2000.

Weitere vier Jahre später, am 1. September 1970, wurde aus dieser einen Stelle (die bis dahin gleichzeitig noch für das Betriebssystem und die Programm-Bibliothek der CD 3300 verantwortlich war) eine eigene Abteilung mit dem Auftrag zur "Entwicklung und Betreuung von Methoden zur Verarbeitung von Textdaten aller Art", um "vor allem den geisteswissenschaftlichen Fachbereichen der Universität Zugang zu dem für sie relativ neuen Hilfsmittel EDV" zu ermöglichen (aus: "Mitteilungen an unsere Benutzer" des ZDV vom September 1970, S. 5).

Schon zwei Jahre zuvor, im Mai 1968, war das Rechenzentrum in "Zentrum für Datenverarbeitung" umbenannt worden. Damit sollte im Namen ausgedrückt werden, daß es seine Aufgabe nicht nur im Umgang mit Zahlen sah - eine Entscheidung, die auch unabhängig von unserem Arbeitsgebiet von der damaligen Weitsicht der Universitätsleitung und der Leitung des Rechenzentrums zeugte: denn die Rolle des Computers hat sich seither in der Tat radikal verändert; heute spricht man kaum mehr von Rechnern oder Rechenzentren, sondern von Informationstechnologie und von Informations- und Kommunikationszentren. Wenn über deren Leistung berichtet wird, sind es nicht mehr die Megaflops, die solche Aussagen prägen, sondern die Statistik der Zugriffe auf Texte, Bilder und Datenbanken, das e-mail-Aufkommen und die Zahl der ans Netz angeschlossenen PCs.

2. Heutiger Stand

Welchen Standort haben unsere seit 1970 institutionalisierten Arbeiten in diesem völlig veränderten wissenschaftlichen und technischen Umfeld?

Die Arbeiten der Abteilung "Literarische und Dokumentarische Datenverarbeitung" des ZDV werden heute häufig mit der Entwicklung von TUSTEP gleichgesetzt. In der Tat ist dieses Werkzeug aus der 1970 definierten Aufgabe hervorgegangen, insbesondere aus der Notwendigkeit, viele Projekte aus praktisch allen geisteswissenschaftlichen Fachgebieten zu betreuen.

Welche Bandbreite an Aufgaben dabei berücksichtigt wurde, läßt sich aus den Protokollen der seit 1973 regelmäßig stattfindenden "Kolloquien über die Anwendung der Elektronischen Datenverarbeitung in den Geisteswissenschaften" an der Universität Tübingen ablesen. Da eine ausführliche Dokumentation dazu im Internet verfügbar ist, möchte ich trotz der runden Zahl 80 auf einen Rückblick auf diese international beachtete Serie von Kolloquien verzichten und auf die Möglichkeit zur Information im WWW verweisen: <http://www.uni-tuebingen.de/zdv/tustep/kolloq.html>. Dies gibt uns etwas mehr Zeit für einen Blick auf die Ursprünge und Grundlagen dieses Werkzeugs.

Die Zeit, in der die Grundkonzepte von TUSTEP entstanden sind, war die Zeit der Großrechner, die mit Lochkarten "gefüttert" wurden. Allein schon deshalb wird TUSTEP gelegentlich als "antiquiert" oder "steinzeitlich" bezeichnet. Für den Anwendungsbereich, für den TUSTEP entwickelt wurde (der

freilich nicht der Consumer-Markt ist, auch nicht der wissenschaftliche Consumer-Markt), sind diese Grundkonzepte aber nach wie vor nicht nur tragfähig, sondern haben sich als unverzichtbar erwiesen. Sie werden auch international als wegweisend für heutige Software-Architektur zur wissenschaftliche Textdatenverarbeitung anerkannt.

So schreibt John Bradley, heute King's College London, der Mitte der 80er Jahre an der Universität Toronto die weit verbreiteten Text Analysis Computing Tools (TACT) entwickelt hat, am 26. September 2000 in der Humanist Discussion List: "TuStep ... provides a very solid set of 'primitives' for the scholarly manipulation of texts. ... Anyone seriously interested in thinking about what a design needs to include in detail would benefit much from examining TuStep"; und am 25. September hat Willard McCarty, der Moderator der Humanist Discussion Group, festgestellt: "Certainly TUSTEP is an example of the kind of working environment I was asking about, and I'd hazard to say that no work along these lines could afford to ignore it." Michael Sperberg-McQueen, einer der Verfasser der TEI-Guidelines und heute Mitarbeiter im WWW-Consortium und dort maßgeblich an der Entwicklung von XML beteiligt, charakterisiert es in einem Beitrag zu dem 1996 erschienen Buch "The Literary Text in the Digital Age" als "unusually well designed".

Diese Anerkennung gilt nicht nur der Entwicklungsmannschaft, sondern auch den Kolloquiumsteilnehmern, von denen viele zur Entwicklung diese Konzepts beigetragen haben. Denn TUSTEP ist nicht am berühmten grünen Tisch entstanden, sondern in enger Kooperation mit vielen anspruchsvollen Projekten aus praktisch allen geisteswissenschaftlichen Fächern.

3. Die Voraussetzungen für die Entwicklung von TUSTEP

Dies interdisziplinäre Arbeit - und die Tatsache, daß TUSTEP in einer Zeit entstanden ist, in der die verfügbaren technischen Mittel für solche Aufgaben nicht ausgelegt und vielfach absolut unzureichend waren - ist die Basis für die heutige Leistungsfähigkeit von TUSTEP.

Wir hatten in der Tat, so absurd dies klingen mag, geradezu ideale Voraussetzungen bei der Entwicklung: unsere Phantasie war nicht eingeschränkt dadurch, daß wir uns an Leistungsmerkmalen vorhandener Software orientieren mußten - es gab ja keine Software für unser Feld. Im Unterschied zu heute, wo jeder, der schon einmal ein Office-Paket in der Hand gehabt hat, auch beurteilen zu können glaubt, welches Werkzeug eine Wissenschaft braucht, die professionell mit Texten umgeht. Ich meine es ernst, daß wir es gut hatten, und bin überzeugt, daß es heute nicht mehr gelingen würde, ein System wie TUSTEP zu entwickeln. Zu dominant wären die Vorgaben, die z.B. die (für viele wissenschaftlichen Zwecke noch immer unzureichenden) Möglichkeiten der Zeichenkodierung von Unicode bedeuten, oder die Vorgaben zur Textauszeichnung,

die mit Standards wie XML und, auf unserem Gebiet, mit den Empfehlungen der Text Encoding Initiative (TEI) heute existieren.

Lassen Sie mich an zwei Beispielen verdeutlichen, wie wir zu den Konzepten kamen, die noch heute TUSTEP ausmachen.

3.1. Hexameter-Analyse und ihre Folgen

Im Jahr 1966 hatte ich begonnen, lateinische Hexameter metrisch zu analysieren, mit dem Ziel, die Materialien, die E. Norden in den stilistisch-metrischen Anhängen seines berühmten Kommentar zum 6. Buch der Aeneis gesammelt hatte, auf eine breitere Grundlage zu stellen und ihnen moderne Texteditionen zugrunde zu legen. Programme, die die Silbenquantitäten und das metrische Schema eines jeden Verses feststellten und anschließend die entsprechenden Materialien (wie die Statistik und die Stellenangabe zur Position der Wortgrenzen im Vers) zusammenstellen, habe ich in FORTRAN geschrieben, unter Verwendung der vom Deutschen Rechenzentrum in Darmstadt entwickelten Unterprogramme, die eine Zeichen- und Zeichenkettenverarbeitung mit FORTRAN erst ermöglichten.

Enttäuscht von dem Ausbleiben des großen Echos, das ich als junger Wissenschaftler für die in Schnelldrucker-Listen vorgelegten Ergebnisse bei meinen philologischen Kollegen erwartet hatte, beschloß ich, das Material zu publizieren. Eine konventionelle Publikation kam aus zwei Gründen nicht in Frage: wegen der Schwierigkeit des Satzes wäre sie für mich unbezahlbar gewesen (man denke allein an den Aufwand, den der Satz der Längen- und Kürzenbezeichnungen über den einzelnen Vokalen gemacht hätten); das Korrigieren der Setzer-Fehler in diesem Material, bei dem es auf jedes Zeichen und jede Ziffer ankommt, wäre ein ebenso langwieriger und teurer wie selbst wieder fehleranfälliger Prozeß gewesen.

Das Satzprogramm

Also mußte ein neuer Weg gesucht werden. 1968 erfuhr ich im Rahmen der Vorarbeiten zur Vulgata-Konkordanz, zu der ich gleich noch komme, daß im Lux-Bildstudio in Neu-Isenburg mit dem DIGISET 50T1 eine über Computer steuerbare, mit Kathodenstrahlröhre arbeitende Lichtsetzanlage zugänglich war.

Ich begab mich also dorthin, ließ mir die Funktionsweise der Anlage erläutern und eine Beschreibung aushändigen und machte mich an die Arbeit, ein Programm zu schreiben, das meine Daten in eine vom Buchdruck gewohnte Form brachte. Anfang 1970 konnte ich die "Metrischen Analysen zur Ars Poetica des Horaz" dann für 1.146,30 DM belichten.

Damit war der Vorläufer des heutigen TUSTEP-Satzprogramms geboren - und damit einer der TUSTEP-Bausteine, mit deren Hilfe es gelang, auch Philo-

logen von der Nützlichkeit der EDV zu überzeugen. Der Vorteil, den es bedeutete, nie mehr Korrektur lesen zu müssen, wenn man einen Text einmal fehlerfrei vorliegen hat, war einsichtig.

Das Problem des Zeichenvorrats

Auch das Problem des Zeichenvorrats war damit prinzipiell gelöst. Im Computer waren damals 6-bit-Codes für die Zeichendarstellung üblich; dies reichte noch nicht einmal zur Unterscheidung von Groß- und Kleinbuchstaben, die wir deshalb durch zusätzliche Zeichen kodieren mußten - von Umlauten, metrischen Symbolen oder den in den Apparat-Einträgen der Vulgata-Konkordanz benötigten Frakturbuchstaben und weiteren Sonderzeichen ganz zu schweigen. Es war somit von vornherein klar, daß wir nach Mitteln suchen mußten, die vorgegebenen Grenzen für die Darstellung zu umgehen. Für die Ausgabe, die ohne solche Grenzen nur über Satzbelichter möglich war, konnten und wollten wir uns keinerlei Grenzen auferlegen, was Zeichenvorrat, Schriftarten, Akzent-Buchstaben-Kombinationen etc. angeht. Dies führte zu der immer noch unübertroffenen effizienten Art der Zeichenkodierung in TUSTEP, die es u.a. erlaubt, beliebige Akzent-Buchstaben-Kombinationen in jeder von TUSTEP unterstützten Sprache zu kodieren, und zwar auf jedem beliebigen Eingabemedium.

Für das Griechische beispielsweise schreiben wir einen Umschaltcode (in diesem Fall #g+) vor den griechischen Teil und benutzen die lateinische Tastatur so, als wäre es die Tastatur einer griechischen Schreibmaschine. "Language tagging" nennt man das heute. Für Akzente und andere Diakritica benutzen wir ein Verfahren, das man inzwischen mit "combining diacritics" bezeichnet: jedes Diacriticum kann mit jedem Buchstaben kombiniert werden. Kodiert wird dies mit einem einleitenden "%", dahinter folgt ein Zeichen, das diesem Diacriticum möglichst ähnlich sieht (z.B. der Schrägstrich für den Akut), dahinter der Buchstabe. Mit TUSTEP können so auch Zeichen dargestellt werden, für die bisher auch Unicode keine Codeposition außerhalb der "private use area" vorsieht. Beim Export nach Unicode, das von TUSTEP als Code für den Datenaustausch unterstützt wird, lassen wir für Zeichen wie z.B. ein übersetztes e, für die UNICODE noch nichts vorsieht, dann eben die TUSTEP-Codierung stehen und wandeln diese in Unicode-Zeichen um. - Dank der Aktivitäten von Herrn Küster in internationalen Normierungsgremien sind im nächsten Release des Unicode-Standards hier Erweiterungen vorgesehen.

3.2. Vulgata-Konkordanz und einige Grundfunktionen

Ich muß noch einmal auf die Anfänge zurückkommen. Denn im Januar 1967 war mit der Mitarbeit an der Vulgata-Konkordanz ein weiteres Projekt begonnen worden, an dem wir für die Entwicklung von TUSTEP viel gelernt haben.

Das automatische Vergleichen von Texten

Ich möchte dies an zwei Teilaufgaben erläutern. Zunächst mußte der Text der Vulgata möglichst fehlerfrei maschinenlesbar gemacht werden. Da Korrekturlesen zu mühsam und zu fehleranfällig ist, wurde beschlossen, den Text 2mal abzuschreiben und per Programm zu vergleichen. Das Ergebnis wurde in Form einer Liste ausgegeben, in der abweichende Stellen markiert waren. Die Korrektur erfolgte maschinell durch Eingabe der Satznummer, falls bei Abweichungen nicht die jeweils als erste ausgedruckte Zeile die richtige war. Nur etwa 80 Zeilen, in denen beide Abschriften Fehler aufwiesen, mußten dabei ganz neu geschrieben werden. Innerhalb von acht Monaten wurde so eine fehlerfreie Version des Vulgata-Textes erstellt - und nebenbei war damit der automatische Textvergleich als Ersatz für das Korrekturlesen an einem größeren Corpus (gut 100.000 Zeilen) erprobt.

Die Erfahrungen vor allem aus diesem Projekt flossen auch in das Design des TUSTEP-Dateiformats ein: Speicherung der Daten in Datensätzen, deren jeder eine Nummer hat. Diese Eigenschaft hat sich als äußerst fruchtbar erwiesen. Darauf beruht beispielsweise einer der wesentlichen Vorteile des TUSTEP-Satzprogramms vor anderen Satzprogrammen: parallel zur eigentlichen Satzausgabe wird eine zweite Datei, die Ziel-Datei, erzeugt, in die der Text aus der Quelldatei unverändert, mit allen Steueranweisungen, aber mit der neuen Seiten- und Zeileneinteilung und entsprechender Nummerierung, ausgegeben wird. Auf diese Weise wird die beim Satzvorgang entstehende Zusatzinformation - nämlich die neue Anordnung des Textes und die zugehörigen Referenzen - völlig selbstverständlich für weitere Verarbeitungsschritte, z.B. für die Registererstellung, automatisch faßbar.

Das automatische Vergleichen haben wir auch bei der Datenerfassung für die Hexameteranalyse genutzt, dabei aber die beiden Abschriften von zwei verschiedenen Ausgaben (für Vergil: Sabbadini und Mynors) anfertigen lassen. Ein erster Vergleich führte zu einer Liste von Abweichungen, die einerseits die Schreibfehler der Hilfskräfte enthielten, andererseits aber auch die Abweichungen der zugrundegelegten Ausgaben. In einem ersten Korrekturgang wurden beide Abschriften nach der Textvorlage korrigiert. Ein zweiter automatischer Vergleich der korrigierten Daten enthielt dann nur noch die Unterschiede zwischen den beiden Ausgaben - darunter auch einige offensichtliche Fehler in den Ausgaben, die wir im Fall von Mynors dem Herausgeber für die Berücksichtigung in weiteren Auflagen mitteilen konnten. Dies war für uns die erste Anwendung des Textvergleichs für das Erheben von Textvarianten, das bei der Vorbereitung kritischer Editionen eine wesentliche Rolle spielt. Das Design des Programms VERGLEICHE, das seit Herbst 1975 im wesentlichen seine heutige Form hat, profitierte natürlich von diesen Erfahrungen.

Schritte zur Lemmatisierung

Zurück zur Vulgata. Für die Konkordanz mußten die flektierten Wortformen auf die grammatikalische Grundform zurückgeführt werden. Für diesen Arbeitsgang, die sog. Lemmatisierung, konnten wir uns des "Lexicon Electronicum Latinum" bedienen, das Roberto Busa für die Arbeit an seinem Index Thomisticus erstellt hatte.

Das ging so: zunächst wurde der Text in Wortformen zerlegt. Jede Wortform wurde mit der Satznummer und der Wortnummer der Zeile versehen, aus der sie gewonnen wurde, und alphabetisch sortiert. Das Lexikon wurde nach den gleichen Sortierregeln sortiert und anschließend in die sortierten Wortformen eingemischt, immer zuerst der Eintrag aus dem Lexikon, dahinter die gleichen Formen aus der Vulgata, zu denen in einem weiteren Schritt die zugehörigen Grundformen aus dem (davor stehenden) Lexikon-Eintrag ergänzt wurde. Da es flektierte Formen gibt, die mehreren Grundformen zugeordnet werden können, haben wir im Schnitt 2,5 Grundformen für jede in der Vulgata vorkommende Wortform erhalten.

Jetzt wurde die um die Grundformen ergänzten Wortformen in die Textreihenfolge zurücksortiert; zur leichteren Kontrolle wurden die zugehörigen Textzeilen mit ausgedruckt. Diese Listen mußte nun von Hand durchgesehen und korrigiert werden. Die Korrektur bestand in der Regel aus der Auswahl der Zeile, die die für die jeweilige Stelle zutreffende Grundform enthielt; in wenigen Fällen, vor allem bei flektierten Eigennamen, enthielt das Lexikon keinen Eintrag; hier mußte die Grundform auf dem Korrekturweg ergänzt werden. Dann wurde das Material wieder sortiert, diesmal nach der lexikalischen Grundform. Für die anschließende Erstellung der Konkordanz wurde beim Wechsel der Grundform im sortierten Material eine neue Zwischenüberschrift erzeugt; die Stellenangaben (noch immer in Form von Zeilennummern) wurden dazu benutzt, um aus der "Volltext-Datenbank" (so würde man heute sagen) den jeweiligen Text zu holen und samt Stellenangabe darunter auszugeben. Das ganze wurde angereichert mit Steuer codes für das Satzprogramm, das anschließend die fertigen Seiten zur Belichtung auf dem DIGISET ausgab. (Das ganze geschah auf einer CD 3300 mit 112 K Worten = 336 K byte Arbeitsspeicher und 48 MB Plattenplatz.)

4. Von der Programmierung zum Programmpaket

Für die beiden genannten Projekte (und noch für einige weitere) waren jeweils maßgeschneiderte Programme in FORTRAN mit den bereits erwähnten Unterprogrammen geschrieben worden. Als die Zahl der Projekte zunahm, war diese Art zu arbeiten nicht mehr durchführbar: es mußte eine Möglichkeit gefunden werden, die dem Anwender erlaubt, ohne Programmierkenntnisse Lösungen für seine Aufgaben selbst zusammenstellen zu können. Vorgefertigte Lösungen,

die es Anfang der 70er Jahre durchaus schon gab (z.B. das 1967 in England entstandene Programm COCOA für "Word Count and Concordance Generation on Atlas"), waren dafür zu starr.

Über solche Probleme mußte ich seit November 1970 nicht mehr allein nachdenken. Und damit bin ich beim vierten Jubiläum (oder genauer gesagt: bei einer wesentlichen Voraussetzung dafür, daß wir den in der Einladung zuerst genannten Anlaß in diesem großen Kreis feiern können): Kuno Schälkle ist seit 30 Jahren und 17 Tagen mit dabei (wenn man von vier Wochen absieht, die er zuvor als Hilfskraft tätig war) und sorgt bei der Planung neuer Leistungen in TUSTEP und bei deren Umsetzung für das reibungslose Zusammenspiel der einzelnen Teile und deren nahtlose Intergration in das Ganze, für deren Zuverlässigkeit und Leistungsfähigkeit, für die Portabilität über Rechnergenerationen und Betriebssysteme hinweg und nicht zuletzt für die Präzision der Beschreibungen.

Wie gesagt: wir kamen zum dem Schluß, daß andere Lösungen gebraucht werden als projektbezogenes Programmieren auf der einen Seite und vorgefertigte Anwendungen auf der anderen Seite. Um wirklich flexibel auf unterschiedliche Anforderungen reagieren zu können, werden Bausteine gebraucht, die relativ elementare Grundfunktionen einzeln bereitstellen, für die Index- und Konkordanzarbeit also z.B. je einen eigenen Baustein für das Zerlegen des Textes, für das Hinzufügen von Sortierschlüsseln, für die Sortierung selbst, ggf. für das Einmischen anderer Daten, für das Zusammenfassen von Zusammengehörigem nach der Sortierung, und schließlich für das Formatieren und Drucken des fertigen Index. Die konsequente Modularität von TUSTEP mit der praktisch beliebigen Kombinierbarkeit der einzelnen Bausteine folgt aus diesen Überlegungen.

Dieses Konzept trägt zwei gegensätzlichen Anforderungen Rechnung: einerseits muß der Anwender die Möglichkeit (oder besser: gar keine andere Wahl) haben, jeden einzelnen Schritt eines Lösungswegs selbst zu definieren; nur so weiß er wirklich, was sein Programm tut, und nur so kann er die Verantwortung für seine Ergebnisse übernehmen. Andererseits müssen die Elementarschritte der Textdatenverarbeitung die vielen technischen Details von hardware, Betriebssystemen und Programmiersprachen vor dem Anwender verbergen. Nur so bleibt der Lösungsweg überschaubar und beherrschbar. Was dabei herausgekommen ist, ist eine Art Skript-Sprache mit sehr leistungsfähigen Grundfunktionen für den wissenschaftlichen Umgang mit Texten.

Natürlich ist ein so aufgebautes System schwerer zu erlernen und zu bedienen als das, was heute auf graphischen Oberflächen an Standard-Lösungen angeboten wird. Diese Eigenschaft, die TUSTEP oft vorgeworfen wird (und die oft mit schmückenden Beiwörtern wie "antiquiert" oder "steinzeitlich" bedacht wird), ist systembedingt. Ich zitiere hierzu noch einmal den Humanist-Beitrag von John Bradley vom 26. September 2000: "Any tool meant to support activities as diverse as those that turn up in humanities text-based computing cannot

possibly be trivial to learn or use. The level of professionalism and commitment required for a full use of TUSTEP is, I think, roughly comparable to that required to learn to work with, say, Perl, or (I think) Smalltalk and text-oriented Smalltalk objects”.

Zumindest für die älteren unter den Kolloquiumsteilnehmern ist dies alles nicht neu. Denn nicht nur die ehemaligen und die gegenwärtigen Mitarbeiter der Abteilung - neben den bereits genannten Herren Küster und Schälkle seien wenigstens die derzeitigen Mitarbeiter, Herr Fuchs, Herr Dr. Kopp und Herr Kottke auch namentlich genannt - haben in je verschiedener, aber im ganzen unverzichtbarer Weise durch Anregungen, Beratung, Kurse, Projektbetreuung, Programmierung oder Übernahme organisatorischer Aufgaben zum Gelingen des Ganzen und zur Stabilität über die Jahre beigetragen. Darüber hinaus haben viele Anwender mit zum Teil erheblichem Einsatz daran mitgearbeitet, daß dieses Werkzeug zur Verfügung steht, und tragen durch Hinweise auf Fehler und durch neue Anregungen weiterhin dazu bei; auch dafür in unser aller Namen herzlichen Dank - und auch dafür, daß sie immer wieder auch öffentlich äußern, was sie an diesem Werkzeug haben: manchmal in recht knappen, aber umso überzeugteren Worten wie: “unter Verwendung der formidablen TUSTEP-Satzprogramme” (Wolfgang Schenkel 1997 im Nachwort seiner Einführung in die klassisch-ägyptische Sprache und Schrift), oder “Märchenhaftes Hilfsmittel dabei war das Tübinger System von Textverarbeitungsprogrammen (TUSTEP)” (Friedrich Wolfzettel im Vorwort des 1993 erschienenen Bandes “Fiktionalität im Artusroman”).

Eines der schönsten Komplimente für TUSTEP habe ich auf der diesjährigen ITUG-Tagung in Erfurt von Prof. Gleßgen aus Straßburg gehört, der sagte: “TUSTEP mogelt nicht”. In der Tat halte ich die Möglichkeit, sehr kontrolliert arbeiten zu können, jeden Schritt selbst definieren zu können und zu müssen, sich darauf verlassen zu können, daß das Ergebnis wirklich aus den selbst niedergeschriebenen Regeln kommt (die vielleicht nicht immer mit dem übereinstimmen, was man eigentlich formulieren wollte) - diese Eigenschaften einer Software halte ich in der Tat für eine unabdingbare Voraussetzung für wissenschaftlichen Umgang mit Texten. Prof. Castrillo aus Burgos hat dies nur anders formuliert, als er den ITUG-Kongress 1999 in Burgos unter das Motto gestellt hat: “TUSTEP educa” - erzieht zum klaren Denken, weil es zwingt, ein Problem in Teilschritte zu zerlegen, die im Detail durchschaubar sind.

Wie wichtig nicht nur meinen Mitarbeitern und mir diese Eigenschaften sind, die vielen heute durchaus unmodern erscheinen, zeigt die Initiative, die die ITUG auf der schon genannten Tagung im Oktober dieses Jahres in Erfurt gestartet hat, um für die Zukunft von TUSTEP eine noch breitere Grundlage zu schaffen.