25 Jahre ITUG: Rückblick und Ausblick

(zu: »ITUG 25.pptx« vom 5.10.2018 mit Eingangsstatements von Wilhelm Ott; die Zahlen in runden Klammern verweisen auf die Folien-Nummern)

(1) Die Organisatoren unserer Jubiläums-Tagung haben mich gebeten, zu Beginn unserer Diskussionsrunde einen kurzen Rückblick auf die Zeit vor der Gründung der International TUSTEP Users Group zu geben, insbesondere auch über die Geschichte und die Vorgeschichte von TUSTEP und die Ideen, die uns bei der Entwicklung geleitet haben.

(2) Ist es Zufall, dass diese Vorgeschichte vor 52 Jahren begann? Sparsam wie wir sind, (3) brauchen wir noch nicht einmal andere Ziffern zu benutzen als für das ITUG-Jubiläum, um die Zeit zu benennen, die verflossen ist, seit die ersten Ideen und Bausteine für das entstanden sind, was vor (4) 40 Jahren den Namen TUSTEP bekam. Organisatorische Voraussetzung war, dass sich die die Universität Tübingen vor 52 Jahren entschlossen hat, für die Unterstützung geisteswissenschaftliche Forschungsarbeit durch EDV an ihrem Rechenzentrum eine Stelle für einen wissenschaftichen Mitarbeiter zu schaffen.

(5) Lassen Sie uns zunächst einen kurzen Blick auf einige weitere wichtige Stationen in die Entwicklung von TUSTEP werfen.

1970 enthält 3 wichtige Daten:

– das Satzprogramm, das eine fehlerfreie und kostengünstige Publikation der Ergebnisse möglicherweise langjähriger Arbeit bietet, hilft, Interesse für den Computer-Einsatz in den Geisteswissenschaften zu wecken.

– Aus der einen Mitarbeiterstelle für die DV in den Geisteswissenschaften wurde die Abeilung LDDV, und mit Herrn Schälkle konnte ein Mitarbeiter gewonnen werden, ohne den die Arbeit der nächsten (inzwischen 48) Jahre nicht denkbar gewesen wäre.

Die meisten von Ihnen kennen die Tübinger Kolloquien zur EDV in den Geisteswssenschaften, deren erstes im Herbst 1973 stattfand.

1976 war es so weit, dass die Tübinger Programme auch außerhalb Tübingens Interesse fanden.

Vor 40 Jahren, beim 14. Kolloquium im Februar 1978, erhielt das Tübinger System von Textverarbeitungsprogrammen seinen Namen.

1985 war mit dem Start des Forschungsschwerpunkts 08 das wohl wichtigste Datum für die Entwicklung von TUSTEP zu dem, was es heute ist: wir konnten nicht nur mit erweiterter Mannschaft die Entwicklung weiter vorantreiben; genau so wichtig oder noch wichtiger war, dass dies als Forschungsschwerpunkt des Landes mit der Auflage verbunden war, diese Programme zumindest landesweit an den anderen Hochschulen verfügbar zu machen. Damit war eine wichtige bürokratische Hürde für die Verbreitung von TUSTEP weggefallen.

(6) Über das Jahr 1993 mit der Gründung der ITUG und über die wichtige Rolle, die die ITUG nicht nur für die Unterstützung bei der Nutzung von TUSTEP, sondern für die Zukunft von TUSTEP selbst spielte und spielt, werden wir gleich von den Personen hören, die diese Gruppe initiiert und geleitet haben.

Ich möchte mich auf ein paar wichtige Daten zur Entwicklung von TUSTEP seit 1993 beschränken.

Seit 1994 können Abbildungen in die Satz-Ausgabe eingebunden werden.

Inzwischen ist die Text Encoding Initiative und ihr Datenformat zum Standard geworden. Wir hatten das Glück, in Tübingen insgesamt 4 Workshops zu TEI organisieren zu können, die von einem der Mitbegründer und Hauptverantwortlichen für diesen Standard angeboten wurden. TUSTEP war eines der ersten Programme, mit denen TEI-kodierte Daten auch zum Satz befördert werden konnten.

Im dritten dieser Workshops im Februar 1997 hat Michael Sperberg-McQueen davon berichtet, dass SGML durch XML abgelöst wird. Mit der Version November 1997 konnte TUSTEP dann auch XML als Auszeichnungssprache für #SATZ verarbeiten, also noch, bevor XML 1998 als W3C recommendation verbschiedet wurde.

Im Januar 1998 startete eine Serie von 19 TUSTEP-Workshops in Blaubeuren.

2003 wurde mit meiner Verabschiedung in den Ruhestand auch die Abteilung LDDV aufgelöst. Dank der Initiative der ITUG und der Bereitschaft von aus-

wärtigen wissenschaftlichen Einrichtungen, als Kooperationspartner auch finanziell zur Weiterentwicklung von TUSTEP beizutragen, konnte die Weiterarbeit von Herrn Schälkle an TUSTEP und die Bereeitstellung der notwendigen Infrastruktur (Webserver, Server für Entwicklung und Pflege der Programme) durch das Tübinger ZDV gesichert werden.

2009 startet der Versuch, TUSTEP mit einer XML-Oberfläche einer an XML-Werkzeuge gewohnten Nutzerschaft zugänglich zu machen.

Ein weiterer wichtiger Schritt für die Zukunft von TUSTEP war, dass 2011 das Rechtsamt der Universität Tübingen zugestimmt hat, dass TUSTEP als Open Source Produkt weitergegeben werden kann.

2013: Zu TUSTEP-Wiki wollte ich wenigstens das Datum nennen.

Das gedruckte Handbuch zur Version 2016 erwähne ich vor allem deshalb, (7) weil hier ein Stück Entwicklungsgeschichte optisch deutlich wird (die Aufnahme zeigt die gedruckten Handbücher von 1985 – 1987 – 1989 – 1993 – 2001 – 2008 – 2016).

Lassen Sie mich noch einmal ganz zurück zu den Anfängen springen und ein paar zusätzliche Anmerkungen nachtragen.

Die Vorarbeiten an den ersten Bausteinen begannen nicht mit meiner Anstellung am Rechenzentrum, sondern ein halbes Jahr vorher mit ersten Programmierübungen zur metrischen Analyse lateinischer Hexameter-Dichtung, die ich bei der Teilnahme an einem Programmierkurs im Deutschen Rechenzentrum (DRZ) in Darmstadt machte.

(8) Das DRZ war Mitte der 1960er Jahre das Rechenzentrum für die deutschen Universitäten, offiziell eröffnet im Juni 1963 mit einer IBM 7090, von der wir hier vor allem die meisten der insgesamt 13 Magnetbandstationen sehen (Plattenspeicher gab es noch nicht).

Im DRZ gab es bereits eine Abteilung Nichtnumerik. (9) Diese hatte bereits einen Satz von (Assembler-)Unterprogrammen entwickelt, um die Programmiersprache FORTRAN außer für den Umgang mit Zahlen und Formeln auch für die Arbeit an Texten zugänglich zu machen, und die auch Programmierkurse für Geisteswissenschaftler anbot.

(10) Programmierung und Dateneingabe erfolgte über Lochkarten mit einem Zeichenvorrat von 48 BCD-Zeichen (später 64 EBCDIC-Zeichen). Dies bedeutete, dass man sogar den Unterschied zwischen Groß- und Kleinschreibung durch Zusatzzeichen markieren musste (hier durch den * vor dem TROJAE in der zweiten Zeile).

(11) Es war die Absicht des Philologischen Seminars der Universität Tübingen, mir die Weiterarbeit (12) an meinen Hexameterstudien – und damit den Beginn der Computer-Unterstützung weiterer geisteswissenschaftlicher Forschungsaufgaben – zu ermöglichen, die dazu führte, dass an dem gerade interfakultär gewordenen Rechenzentrum am 1.10.1966 eine Stelle für diese Aufgaben geschaffen wurde. (13) Mitte 1967 erhielt das Tübinger RZ mit der Control Data 3200 mit 32 K 24-Bit-Wörtern ersten Rechner der 3. Generation. Hier (14) ein Blick mehr als 10 Jahre später in den Benutzerraum.

Um in Tübingen weiterarbeiten zu können, musste ich als erstes ein zu den Darmstädter Unterprogrammen aufruf-kompatibles Paket von Unterprogrammen zur Zeichenverarbeitung für die Tübinger Anlage schreiben. Das war die Basis für die Programmierung für die neuen Projekte, die auf uns zukamen. Das Paket wurde später überarbeitet, vom Assembler auf C umgestellt und wird noch heute in vielen TUSTEP-Bausteinen benutzt.

(15) Auf das erste größere auswärtige Projekt, die Vulgata-Konkordanz, die von P. Bonifatius Fischer vom Vetus-Latina-Institut aus einer von ihm mit herausgegebenen neuen Vulgata-Edition erstellt wurde, möchte ich etwas näher eingehen, weil wir daran viel für die Architektur von TUSTEP gelernt haben.

(16) Diese Konkordanz sollte natürlich eine lemmatisierte Konkordanz sein, d.h. die Belege sollten unter einer Überschrift stehen, die die lexikalische Grundform der im Kontext stehenden Wortformen enthält.

(17) Zwar gab es auch damals schon Programme zur Index- und Konkordanzerstellung, wie das Programm INDEX des DRZ, mit dem auch eine (18) »Stellenkonkordanz« erstellt werden konnte, oder (19) COCOA, das »Word Count and Concordance Generation on Atlas«, aus dem später das Oxford Concordance Program von Susan Hockey und Ian Marriot hervorging. Mit Programmen dieser Art war es weder möglich, eine lemmatisierte Konkordanz zu erstellen, noch Teile davon für einzelne der notwendigen Arbeitsschritte zu nutzen: es waren Black Boxes, die neben dem zu indizierenden Text noch

Parameter für dessen Gestaltung, aber keine Möglichkeiten zur Interaktion mit anderen Programmen oder Programmteilen vorsahen.

Für die Lemmatisierung konnten wir uns des »Lexicon Electronicum Latinum« bedienen, das Roberto Busa für die Arbeit an seinem Index Thomisticus erstellt hatte.

(20) Zuerst wurde der Text jeder als Kontext auszugebenden Zeile in einzelne Wörter (besser: Wortformen) zerlegt. Jede Wortform wurde mit der Nummer der Zeile und der laufenden Nummer des Wortes in der Zeile versehen, aus der sie gewonnen wurde, und (21) alphabetisch sortiert.

(22) Das LEL wurde nach den gleichen Sortierregeln sortiert und in die sortierten Wortformen eingemischt, immer zuerst der Eintrag aus dem Lexikon, dahinter die identischen Wortformen aus der Vulgata, (23) zu denen in einem weiteren Schritt die zugehörigen Grundformen aus dem 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.

(24) Jetzt wurden die um die Grundformen ergänzten Wortformen in die Textreihenfolge zurücksortiert und (25) zusammen mit ihre Kontext ausgedruckt. Nur so konnte festgestellt werden, um welches Lemma es sich jeweils handelt. Bei Wortformen, die mehreren Lemmata zugeordnet waren, musste dann die Zeile mit dem jeweils gültigen Lemma ausgewählt werden, falls dieses nicht das erste einer Wortform zugeordnete Lemma war. Für manche Formen, vor allem für Eigennamen, enthielt das Lexikon keinen Eintrag; hier mußte bei flektierten Formen die Grundform auf dem Korrekturweg ergänzt werden. (26) Dann wurden die Einträge wieder sortiert, diesmal nach dem Lemma.

(27) Für die anschließende Erstellung der Konkordanz wurde beim Wechsel der Grundform eine entsprechende 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. (28) Das ganze wurde angereichert mit Steuercodes für das Satzprogramm, (29) das anschließend die fertigen Seiten zur Belichtung auf dem DIGISET ausgab.

(30) Bevor die Arbeit an der Konkordanz begann, musste der Text erst einmal maschinenlesbar vorliegen. Bonifatius Fischer, der Bearbeiter, sagte, er schreibe sehr viel genauer als er Korrektur lesen könne. Deshalb schlug ich vor, dass er den Text zweimal abschreibt und wir die beiden Abschriften dann vergleichen. Sie sehen hier eine Zeilendruckder-Liste aus dem Jahr 1967, die mehr oder weniger stark an das Vergleichsprotokoll von heute erinnert.

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 geschreiben 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 wir eine zweite Datei, die Ziel-Datei, erzeugt, in die der Text aus der Quelldatei unverändert, mit allen XML-Tags und anderen 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 zugänglich, z.B. für die Registererstellung oder die Auflösung seitenbezogener Querverweise.

(31) Für das, was wir daraus gelernt hatten, möchte ich aus dem Protokoll des 80. Kolloquiums vom 18.11.2000 zitieren:

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.

Andererseits kommen bei aller Verschiedenheit der einzelwissenschaftlichen Aufgabenstellungen bestimmte elementare Arbeitsschritte immer wieder vor, nicht nur für die Eingabe und Korrektur der Daten und die Ausgabe in professioneller typographischer Qualität. Auch für die Analyse- und Verarbeitungsschritte gibt es Grundfunktionen, die in jedem Projekt gebraucht werden, dabei aber jeweils dem Einzelprojekt angepaßt und in jeweils unterschiedlicher Abfolge zu Problemlösungen zusammengestellt werden müssen.

Über solche Probleme mußte ich seit November 1970 nicht mehr allein nachdenken. Kuno Schälkle ist seit 48 Jahren mit dabei 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.

(32) Das Ergebnis unserer frühen Erfahrungen und der daraus folgenden Lösung kennen Sie: bis auf KOPIERE, EINFUEGE und NUMERIERE haben wir die hier aufgeführten Bausteine von TUSTEP bzw. deren Vorläufer schon alle genannt in diesem kurzen Rückblick, den ich hiermit schließen möchte.