Zeitschrift: Bulletin CILA : organe de la Commission interuniversitaire suisse de linguistique appliquée

Herausgeber: Commission interuniversitaire suisse de linguistique appliquée

Band: - (1990)

Heft: 52

Artikel: Der computergestützte Sprachunterricht in der Sackgasse?

Autor: Wazel, Gerhard

DOI: https://doi.org/10.5169/seals-977989

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 09.12.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Der computerunterstützte Sprachunterricht in der Sackgasse?

Nach der Euphorie der «Gründerjahre» des computerunterstützten Sprachunterrichts ist Ernüchterung eingetreten. YOUNG 1988, 65) charakterisiert die gegenwärtige Situation durchaus zutreffend, wenn er schreibt, der anfängliche Enthusiasmus für Mikrocomputer als motivierende Hilfen für das Lehren einer Fremdsprache sei wegen der begrenzten Formen von Software, die bisher für das Sprachenlernen zu bekommen sind, weitestgehend verlorengegangen. Einer der führenden Vertreter der Sprachlehrforschung in den USA, St. KRASHEN (1985), schlägt den Lehrern sogar vor, sie sollten all ihre Sprachsoftwaredisketten löschen und sie stattdessen für die Textverarbeitung nutzen. Diese Empfehlung ist zweifellos nicht ganz so ernst gemeint, aber sie enthält durchaus einen rationellen Kern: Die kaum noch überschaubare Masse der von Hobbyinformatikern und -pädagogen produzierten und auf den Markt bzw. in Umlauf gebrachten Computer-Sprachprogramme ist eher geeignet, den Unterricht zu verändern, als ihn zu verbessern; denn diese Programme entsprechen entweder programmtechnisch nicht dem der fortgeschrittenen Hardware adäquaten Standard, oder/und sie basieren auf methodisch und psychologisch überholten Positionen: vornehmlich der grammatikalisierenden Übersetzungsmethode und dem Reiz-Reaktions-Schema des Behaviorismus. Da der Computer einseitig als Instrument der linearen Informationsvermittlung begriffen wird, stehen die lineare tutorielle Einführung lexikalischer, grammatischer etc. Kenntnisse, die Festigung derselben in Gestalt von Drillübungen, Tests und Leseübungen mit relativ anspruchslosen Rückmeldungen im Vordergrund. Es wird nicht erkannt, dass die spezifischen Potenzen des Computers «eher in der Manipulation von komplexen Relationen (liegen), die für das menschliche Hirn zu komplex sind, z.B. der Simulation theoretischer Modelle» bzw. dort, wo es darum geht, «nach Informationen zu suchen, Daten zu sortieren, logische und statistische Relationen und Selektionen von Daten zu bilden und bestimmte inhaltliche Konzepte durch die Verknüpfung von Daten überprüfen zu lassen. . . »; derart organisierte, dem Lerner aus seiner täglichen Praxis bekannte Lernwege gestatten es, «mit Wissen in einer ‚natürlichen' Weise zu experimentieren: Fragen zu stellen, Zusammenhänge herzustellen, zu suchen, beim Suchen zu ‚springen', also sich ähnlich zu verhalten, als würde man mit mehreren Büchern und Karteien gleichzeitig hantieren» (SCHULMEISTER, 1989, 19). Die Forderung nach einem solcherart gestalteten Lernprozess wird gegenwärtig mit Nachdruck gleichermassen von der kognitiven und der Gestaltpsychologie gestellt. Mit ihrer Erfüllung wäre eine Zurückdrängung der

die Weiterentwicklung der Teachware behindernden behavioristischen Lösungen verbunden, die in der Regel die Eigenaktivitäten der Lerner reduzieren sowie den Transfer des Erworbenen auf neue Probleme oder unter geänderten Bedingungen erschweren (vgl. SPADA/OPWIS, 1985, 13ff.) Dass Programme dieser Art, zumal wenn sie in Gestalt von Autorenprogrammen angeboten werden (ausführlicher dazu s. WAZEL, 1988) dennoch auch bei Hochschullehrern sehr beliebt sind, liegt an ihrer leichten Manipulierbarkeit: Ihre Benutzung setzt kaum System- und Programmierkenntnisse voraus. Der Unterrichtendende braucht nur den zu vermittelnden Stoff in relativ kleine Portionen zu unterteilen, die einzelnen Programmschritte folgerichtig nacheinander anzuordnen, Fragen bzw. Aufgaben zu formulieren, Hilfen für die Aufgabenlösungen vorzugeben, ein Wörterbuch einzugeben, richtige bzw. falsche Antworten zu antizipieren und das entsprechende Feedback dazu anzugeben, evtl. Wortwurzeln für einen Orthographiechecker zu benennen, und die Autorensoftware produziert daraus ein Lernprogramm – oft einschliesslich eines Bewertungsteils für die Lernerleistung –, das dann vom Lerner selbständig abgearbeitet werden kann. Die geringen Schwierigkeiten bei der Programmherstellung müssen indessen, wie bereits erwähnt, mit einer doch beträchtlich restringierten didaktischen Konzeption erkauft werden. Diese Einschätzung gilt auch dann noch, wenn, wie neuerdings verschiedentlich zu beobachten, Simulationen und Problemlösungen ermöglichende Unterprogramme einbezogen werden.

Wenn die derzeitige Krise in der Programmproduktion überwunden werden, all die anspruchslosen Grammatik- und Lexikprogramme, die Storyboard-, Choicemaster-, Textmaster-, Gapmaster-, Matchmaster-, Wordstore- und sonstigen inzwischen traditionellen Computerprogramme durch bessere Programme abgelöst werden sollen, ist ein konsequenter Paradigmawechsel notwendig: hin zu offenen Programmen mit relativ grossen Freiheitsgraden, die die Lerner zu einem pluralistischen, nicht-linearen Denken führen.

Wenn dieses entdeckende Lernen erreicht werden soll, sind indessen bei der Programmgestaltung bestimmte Grundsätze zu beachten, etwa die bei MANDL/HUBER (1978, 65) genannten:

- vorgefertigte Lösungen sollen bewusst fehlen,
- es sollen offene Lernsituationen geboten werden,
- der Lerner muss die Chance bekommen, seinen Lernweg selbst zu bestimmen,
- die Problemlösungsprozesse, nicht die Resultate sollten bewertet werden.

Wir glauben, dass einige im Entstehen begriffene bzw. bereits in ersten Fassungen vorliegende maschinelle Verfahren und Softwaretypen gute Chan-

cen dafür eröffnen, eine diesen und unseren an anderer Stelle publizierten Prämissen (vgl. WAZEL 1987, 22ff.) entsprechende Sprachsoftware herstellen zu können.

PARSER

Dazu gehören zunächst meist direkt für Sprachanwendungen (Sprachunterricht, Textverarbeitung, Bibliotheksrecherchen, Mensch-Maschine-Dialog, automatische Übersetzung) gedachte Analyseverfahren, wie diverse PARSER (ausführlicher dazu s. KINDERMANN, 1987, 1ff.; COOK, 1988, 50ff.; MURRAY/MORGENSTERN/FURSTENBERG, 1989, 102ff.; HELLWIG, 1987, 49ff.; SANDERS, 1987, 77ff.; LORITZ, 1987, 51ff.; SANDERS, 1989, 13ff.), worunter wir in Anlehnung an HANDKE (1989, 25) Prozeduren verstehen wollen, bei denen Spracheingaben in ihre funktionalen Einheiten zerlegt werden, wobei ein Lexikon, morphologische Algorithmen sowie ein syntaktisches Regelwerk interagieren. Das Ergebnis der maschinellen Analyse ist eine (graphische oder lineare) Strukturbeschreibung der Spracheingabe (z.B. eines Satzes), die den Ausgangspunkt für den weiteren maschinellen Umgang mit der menschlichen Sprache bildet, d.h. vor allem: die sodann mittels Recognizer zur Entscheidungsfindung darüber genutzt werden kann, ob die betreffende Eingabe grammatisch und/oder semantisch korrekt ist. Die seit etwa einem Jahrzehnt sukzessive ins Blickfeld der linguistischen bzw. informationstheoretischen Forschung gekommenen, meist in den höheren Programmiersprachen LISP oder PROLOG geschriebenen Parser und die Recognizer werden, sobald sie ausgereift sind, die derzeit noch üblichen, umständlichen Listenvergleichsalgorithmen ersetzen, d.h., der Computer braucht dann nicht mehr die Eingabe des Lerners mit einer oftmals fast unendlichen Anzahl antizipierter Richtig- oder Falschantworten zu vergleichen, sondern er setzt die Lernerantwort zu den Regeln seiner internen Grammatik bzw. seines Lexikons ins Verhältnis und ermöglicht so Entscheidungen über die Akzeptabilität der Lernereingabe.

HYPERMEDIA

Relativ hohe Hardwareanforderungen (zumindest einen 640-KB- bzw. 1-MB-Arbeitsspeicher und eine Festplatte) stellt in der Regel auch ein Softwaretyp, der als persönliches Werkzeug zu Verwaltung und Verarbeitung von jeglicher Art Information schlechthin gedacht ist, aber u.E. künftig ausgezeichnet für Sprachlehrzwecke nutzbar gemacht werden kann. Er

wurde wie seine Nachfolger, etwa SuperCard, Course of Action usw., zuerst vor allem für Macintosh-II-Computer unter dem Namen «HyperCard» (HARVEY, 1988) entwickelt und liegt nun in ersten Versionen auch für IBM und Kompatible unter den Bezeichnungen «Guide» und «LinkWay» vor. «HyperCard», «Guide», und «LinkWay» usw. sind auf Theodor NELSONS in den frühen 60er Jahren entwickelten Programmtyp «Hypertext» zurückzuführen. Unter diesem Begriff ist in erster Linie eine mausgesteuerte Text-Bilddatenbank zu verstehen, deren Texte und Bilder mittels indizierter Felder in beliebigen Relationen miteinander zu kombinieren sind. «Hypertext» bedeutet eigentlich ursprünglich «nicht-sequentielles Schreiben». Besonders die neueren Anwendungen gehen indessen weit über dieses Ziel hinaus; denn bereits bei dem Prototyp handelt es sich im Idealfalle um ein komplexes, schnell operierendes, assoziativ verbundenes Netzwerk, das mehrere Anwenderprogramme in sich vereinigt: eine Datenbank, ein Textverarbeitungs- und Zeichenprogramm sowie eine vergleichsweise leicht zu erlernende, gut lesbare, spezifische, objektorientierte, durch externe – etwa in PASCAL, C oder Assembler geschriebene – Befehle zu erweiternde Programmiersprache (bei HyperCard «HyperTalk» genannt). In dieses mit Pull-Down-Menüs, sog. Browsern (Mittel zur schnellen Informationssuche), detailunterdrückenden Filtern, Scrollfunktionen und anderen Tools operierende System können ferner problemlos externe Programme (auf CD-ROMs, Bildplatte, Videorecorder usw.) eingebunden werden. Schliesslich lässt es die Kommunikation mit externen Netzwerken bzw. Datenbanken zu. Man bezeichnet es daher in dieser Erweiterung neuerdings nicht zu Unrecht auch als «*Hypermedia*» (CHIGNELL/LACY, 1988, 17; UNDERWOOD, 1989, 10); denn es eröffnet dem Leser, wenn man so will, Fenster in die Aussenwelt. «Hypermedia» ist nicht zu verwechseln mit «Multimedia», das lediglich den Verbund differenter Medien darstellt. «Hypermedia» hingegen bedeutet eine hochgradig integrierte elektronische Umgebung, die es einem Nutzer ermöglicht, ein grosses Ensemble elektronisch verknüpfter Informationen zu kontrollieren bzw. zu nutzen, die in differenten Formen auftreten: als Videosequenzen, Ton, Text und elektronisch zu ermittelnde Daten (vgl. JAFFE/LYNCH, 1989, 10). Hypertextprogramme sind inzwischen vorrangig für elektronisches Publizieren, Projektmanagement, Systemanalysen, Softwareentwicklung, CAD, die Dokumentation, besonders On-Line-Bibliotheken, öffentliche Informationssysteme, Autorensysteme, kooperative Netzwerk-Arbeitssysteme, Telekonferenzen usw. adaptiert worden (vgl. FIDERIO, 1988, 237f.). Die Zahl der anspruchsvolleren Hypertext-Sprachprogramme ist noch relativ gering (vgl. NAJIMA, 1988, 75ff.).

Hypertext bzw. -media arbeitet auf der Basis der Fenstertechnik (bei dem Prototyp «Guide» müssen die Microsoft-Windows vorher geladen werden); die Fenster bzw. Felder können durch das Anklicken sog. Ikonen (icons) und Knöpfe (buttons) direkt manipuliert bzw. vielfältig miteinander kombiniert werden (sequentiell, taxonomisch und netzförmig). Nach unseren ersten Einsichten eignet sich dieser flexible Prototyp betreffs des Fremdsprachenlernens in erster Linie zur Entwicklung des stillen Lesens und im Zusammenhang damit zur selbständigen Erarbeitung landeskundlicher Kenntnisse durch den Lerner; denn mittels der neuen Technik ist es beispielsweise möglich, Texte und Bilder in all ihren Dimensionen zu erschliessen, d.h., der Lerner kann den Text und das Bild nicht nur sequentiell, sondern an jeder vom Programmierer «aktivierten» Stelle tief hinterfragen, i.e. Relationen ergründen, Hintergrundinformationen abrufen, hinter denen u.U. wiederum Informationen, auch in Gestalt von interaktiven Graphiken, Musik, Bildern, Animationen, Videosequenzen usw., stehen. Da, wie bereits erwähnt, ganz leicht extern erstellte Software zu integrieren ist, können dazwischen Übungsprogramme abgearbeitet, Expertensysteme befragt werden usw.

Ein einfaches Beispiel soll das Gesagte verdeutlichen: Bei der Lektüre eines Werbeprospektes für Studien an der Friedrich-Schiller-Universität Jena erfährt der Schweizer Deutschlehrer aus dem Kanton Neuchâtel oder Jura, der zum inzwischen schon traditionell gewordenen Weiterbildungskurs nach Thürigen fahren soll, dass in dieser «Stapelstadt des Wissens und der Wissenschaft» neben Schiller und Goethe auch andere grosse Philosophen, wie Fichte, Hegel und Schelling; Dichter, wie Johann Christian Günther, Ludwig Tieck, Clemens Brentano, Fritz Reuter, der Slowake Jan Kollar, Gerhart Hauptmann; Naturwissenschaftler, wie Alexander und Wilhelm von Humboldt, Hufeland, Döbereiner und Oken, hier lebten und wirkten und Musiker, wie Robert Schumann, Giacomo Meyerbeer, Max Reger, zu Ehrendoktoren der Universität ernannt wurden.

Mit ziemlicher Sicherheit wird der Leser einige der Namen nicht kennen. Er muss, wenn er seine Neugierde befriedigen will, nach einem Lexikon greifen. Falls ihm die dort zu findenden Informationen nicht ausreichen, wird er sich in einer Bibliothek oder Buchhandlung um eine Universitätsgeschichte oder biographische Darstellungen kümmern müssen. Vielleicht interessiert ihn auch die geographische Lage Jenas, das dortige Klima, Sehenswürdigkeiten, die dort gesprochene Umgangssprache, möglicherweise möchte er ein Werk der genannten Musiker zu Gehör bekommen. Dazu müsste er sich eventuell eine Landkarte, einen Reiseführer, ein Video, eine Schallplatte organisieren.

Bei der Nutzung von Hypertext können all die genannten Wünsche befriedigt werden, ohne dass sich der Leser vom Computer entfernen muss. Er braucht nur mit der Maus das entsprechende Wort anzuklicken bzw. es einzugeben, und schon kann er die gewünschte Detailinformation erhalten, die Landkarte, interaktive stilisierte Graphiken, anschauliche Animationen, einprägsame Bilder aus Vergangenheit und Gegenwart, Sprach- und Musikproben präsentiert bekommen, sofern der Hypertextautor derlei Wünsche antizipiert, die Daten eingegeben und ihre Verbindung durch sog. «Links» gesichert hat. Mitunter bewerkstelligt dies das jeweilige System auch von selbst, was häufig bei grossen Text-Datenbanken mit Kreuzundquerverbindungen der Fall ist. Einschränkend soll aber darauf verwiesen werden, dass die derzeitigen Systeme aufgrund der Notwendigkeit, die Informationen in getrennte, relativ abgeschlossene «Nodes», d.h. etwa: Informationsportionen, zu unterteilen, die Arbeit an grösseren, zusammenhängenden Texten kompliziert gestalten. Sie erfordert vom Anwender eine weitsichtige Grob- und Detailplanung, d.h., sie stellt hohe Ansprüche an seine Fähigkeit zur Antizipation. Werden die Datenfelder nicht nachvollziehbar logisch und übersichtlich verknüpft, kann dies wie bei der Programmiersprache BASIC leicht zu einem Wirrwarr bzw. «Spaghetti-Code» führen. Abgesehen davon macht die komplexe, vernetzte Struktur von Hypermedia-Programmen, wenn sie dem natürlichen menschlichen Lernen nahekommende Lernprozesse ermöglichen soll, ohnehin generell einen beträchtlichen Planungsaufwand erforderlich. Wenn in den einschlägigen Hypertext-Werbeprospekten von der Unkompliziertheit der Programmierung gesprochen wird, dann trifft dies folglich nur insofern zu, dass keine tiefergehenden EDV-Kenntnisse vonnöten, die einzelnen Programmelemente isoliert sind und damit nicht nur vom Originalprogrammierer, sondern auch von anderen potentiellen Autoren modifiziert werden können, mithin auch eine dynamische, evolutionäre Teamprogrammierung leicht zu bewerkstelligen ist; die Schwierigkeiten liegen auf einer anderen Ebene.

Weil in Hypermedia grosse Informationsmengen, Wörterbücher, und Lexika eingeschlossen, mit leichtem Zugriff gespeichert werden können, bietet es auch enorme Möglichkeiten für die Lösung von Problemaufgaben, das Schreiben von Aufsätzen und Gebrauchsformen des wissenschaftlichen Arbeitens, Übersetzungen etc. eingeschlossen. Das Programmsystem dient ergo in diesem Falle als eine dynamische Datenquelle, die vom Lerner je nach Bedarf in verschiedenen Richtungen zu konsultieren ist und die «sowohl willkürliche, assoziativ gewählte Wege, aber auch logische, sachlich-inhaltlich begründete und durch Hypothesen geleitete Pfade durch das Informationsmaterial» ermöglicht (SCHULMEISTER, 1989, 53). Hypermedia fungiert folglich bei entsprechender Konzipierung letztlich

sowohl als Lehr- als auch als Forschungsinstrument, das die oft überdimensionale Recherchezeit zugunsten des Forschens bzw. Lernens verringert. Letztlich hebt Hypermedia die rigide Abgrenzung zwischen Entwicklungssoftware und anwendungsspezifischen Programmen, Programmierer und Anwender auf (vgl. MEYER, 1988, 88); man könnte es auch als eine höhere Stufe eines Autorenprogramms bzw. -systems bezeichnen, das mit Stapeln leerer Karten arbeitet. Diese können vom Anwender je nach Bedarf beschrieben werden. Mit seiner anschaulichen ikonischen Präsentationsweise nähert sich Hypertext der Schaffung des von Laien immer dringlicher geforderten plastischen «Human Interfaces», das zur Emanzipation des Anwenders von den Computerhändlern und Fachinformatikern beiträgt. Von ihnen ist er heute noch oftmals in einem hohen Masse abhängig, bzw. er wird bewusst in dieser Abhängigkeit gehalten. Häufig wird die u.E. indiskutable Auffassung vertreten, mit «dem Computer arbeiten heisse, die Kommandosprache eines Betriebssystems zu beherrschen, in mindestens einer höheren Programmiersprache programmieren zu können und sich die total andere Denkweise einer binären Maschine überstülpen zu lassen» (SCHULMEISTER, 1989, 6). Wenn jedoch der Computer ein die wissenschaftliche Lerntätigkeit effektivierendes Hilfsmittel sein soll, dann müssen die Programme so bedienerfreundlich sein, dass sich der Anwender auf sein fachliches Problem, d.h. den Inhalt, den logischen Ablauf etc., konzentrieren kann und sich nicht als Dilettant im Dschungel einer komplizierten Technologie bewähren muss, die er nicht studiert hat und demnach auch nicht auf dem erforderlichen Niveau beherrschen kann.

Eine erste Vorstellung davon, wie ein entsprechendes Sprachlernprogramm aussehen kann, gibt das Japanischprogramm «Kanji City». Es nutzt die Potenzen von HyperCard zur Integration von Text mit digitalisiertem und synthetisiertem Ton, interaktiven Graphiken und Animationen. Ferner demonstriert das Programm, wie Hypermedia das induktive Lernen durch Versuch–Irrtum-Erforschung einer Simulation von Alltagssituationen entwickeln und kontextualisierte Sprachverwendung befördern kann. «Kanji City» ist auch als Modell für die Einbeziehung extern erstellter Lernprogramme, z.T. in Gestalt sog. Shells, d.h. vom Nutzer bzw. Lehrer mit eigenem Material zu füllender Rahmenprogramme, anzusehen.

In «Kanji City» bewegt sich der Lerner durch eine städtische Umgebung – im konkreten Falle Tokio –, indem er auftretende sprachliche Zeichen liest und entsprechend darauf reagiert. Diese Zeichen treten dem Japanisch-Anfänger sinnvollerweise zunächst in Gestalt von Schildern, Listen, Namen und anderen schriftlichen Formen entgegen, die normalerweise nicht in zusammenhängenden Texten vorkommen. Daneben wird mit Gebrauchsformen, wie Ankündigungen und Anzeigen, gearbeitet. Auch

die konventionellen, auf der Basis von Häufigkeiten zusammengestellten Texte sind so weit wie möglich mittels verschiedener graphischer etc. Mittel multipel kontextualisiert.

Der Lerner wird mit den unterschiedlichsten Themenbereichen vertraut gemacht: Er hat Speisekarten in Restaurants, Schilder in Kaufhallen, Preisaufstellungen und Geschäftskarten zu lesen und je nach Situation darauf zu reagieren. Er wird in eine Diskothek, in Mathematik- und Orthographiestunden, in eine Bank sowie in ein Casino versetzt, kann einen Traumurlaub in Hawaii erleben und hat eine Reihe exakt determinierter sozialer Bewährungssituationen nachzuvollziehen, in denen er sich adäquat zu verhalten hat, wobei ihm Konsequenzen eventuellen Fehlverhaltens vorgeführt werden. Er ist angehalten, den richtigen Zug zu nehmen, die entsprechenden Fahrkarten zu kaufen, an der richtigen Station auszusteigen und mit einer bestimmten Summe Geldes hauszuhalten, sofern er sie nicht beim Sprachspiel im Casino oder bei Unterrichtstests im Programm aufbessern kann. Falls er sprachliche Hilfen, etwa in Form von Übersetzungen ins Englische oder Transkriptionen, in Anspruch nimmt, muss er dafür bezahlen, und die erwähnte Summe verbraucht sich dadurch schneller (ausführlichere Beschreibung s. ASHWORTH/STELOVSKY, 1989, 28ff.).

Die Entwicklung von Hypermedia steht trotz der bereits in den Prototypen vorhandenen erstaunlichen Möglichkeiten noch am Anfang, und die nun einsetzenden Anwendungen und Forschungen (etwa betreffs einer Translate-Funktion, die das Übersetzen der Hypertext-Programmiersprache in diverse Landessprachen besorgt) – nebenbei soll nur erwähnt werden, dass unter der Leitung von J. UNDERWOOD (Western Washington University) in Zusammenarbeit mit CALICO 1989 eine «Hypermedia special interest group» zur gezielten Erforschung und Anwendung von Hypermedia gegründet wurde (OTTO, 1989, 3) – werden zu Verbesserungen, zur Beseitigung von Schwächen, vor allem doch noch vieler Bedienungsschwierigkeiten, und zu weiteren Innovationen führen, die nach unserer Einschätzung in Verbindung mit den Ergebnissen bei der Sprachsynthese- und Parserforschung, der Entwicklung von Expertensystemen und last not least der des Interaktiven Videos (vgl. UNDERWOOD, 1989, 7ff.; 1989, 23ff.) den Sprachunterricht wesentlich in Richtung grösserer Kreativität bzw. Aktivität des Lerners mitprägen werden. Gerade der dynamische Verbund des mit Hypermedia arbeitenden Computers und des Videorecorders mit selbst beschreibbarer Bildplatte wird sich revolutionierend auf den computerunterstützten Fremdsprachenunterricht auswirken, u.a. weil nun ohne die bisherigen Einschränkungen des Bild-Ton-Trägers (etwa langsamer Zugriff beim Einsatz von Videorecordern oder übermässig teure Software bei den nicht selbst beschreib- bzw. löschbaren Bildplatten – vgl. z.B. WEST, 1989,

52; UNDERWOOD, 1989, 8) eine Präsentationsweise des Lehrstoffs möglich wird, die auch die Entwicklung bisher bei Computeranwendungen vernachlässigter Zieltätigkeiten, wie des interaktiven Hör-Seh-Verstehens (vgl. FROMMER, 1989, 52) gestattet, und die Schaffung anspruchsvollerer Programmtypen, z.B. zum Schlussfolgern, Bewerten, Problemlösen, zur Evaluation, Simulation usw., befördert.

EXPERTENSYSTEME

Neben Hypermedia verdient u.E. ein weiterer Softwaretyp Beachtung: die sog. Expertensysteme, die wie die Schachprogramme eng mit der Entwicklung der Künstlichen Intelligenz verbunden sind. Sie werden mitunter als elektronische Handbücher beschrieben, in denen nicht nur umfangreiches und z.T. sehr spezifisches Faktenwissen über einen bestimmten Gegenstandsbereich (Domäne) gespeichert ist, sondern die auch Regeln darüber enthalten, wie dieses Wissen auf Fakten angewendet werden kann. Anders ausgedrückt: Expertensysteme sind Computerprogramme, die Probleme lösen, indem sie gespeicherte Regeln und Fakten benutzen. «Mit ihrer Hilfe lassen sich z.B. Infektionskrankheiten (MYCIN) oder Erkrankungen der Atemwege (PUFF) diagnostizieren, sie erleichtern das Finden von Bodenschätzen oder die Suche nach dem Defekt in einer Maschine oder einem Computer. Nachdem das Programm den Benutzer durch eine Vielzahl vernetzter Fragen über Symptome des zu ermittelnden Gegenstandes oder über die Restriktionen des aktuellen Handlungsrahmens geleitet hat, stellt es auf der Basis der so ermittelten Problembeschreibung durch Rückgriff auf entsprechende Fakten in seiner Wissensbasis eine Diagnose bzw. unterbreitet gegebenenfalls einen Handlungsvorschlag. Expertensysteme sind in den meisten Fällen in PROLOG oder LISP programmiert, und sie bestehen, grob gesagt, aus drei Bestandteilen: der Wissensbasis, die das Problem «erkennt», dem Inferenzprogramm, das entscheidet, wie das Problem zu lösen ist, und dem Nutzerinterface, das Fragen stellt und die Antworten gibt. Die Erstellung eines Expertensystems geschieht in zwei, jeweils noch vielfach untergliederten Stufen. «Zunächst wird eine sog. ‚shell' erstellt, ein Programm, welches Ressourcen zur Erhebung von Wissen über Fakten und Regeln zur Verfügung stellt. In einem zweiten Schritt wird es mit Expertenwissen über eine bestimmte Domäne gespeist.» (BIRKHAN, 1989, 158f.)

Viele Expertensysteme sind so angelegt, dass der Fachmann nach dem vorgegebenen Muster ein Expertensystem für sein spezielles Gebiet schaffen kann, das einem Lerner, allgemein gesagt: einem Benutzer eine schnelle, selbständige Orientierung und Problemfindung ermöglicht. Bei der Arbeit

mit dem von W. Hapgood (WAYLAND, MA) entwickelten, in PASCAL und Assembler geschriebenen, auf einer 5¼-Zoll-Diskette gespeicherten Expertensystem 1st-Class (256 K RAM notwendig) hat er z.B. folgendermassen vorzugehen:

- Er definiert die Begriffe der möglichen Resultate und der diese Resultate bestimmenden Begriffe;
- danach gibt er einige Beispiele in ein Spreadsheet (bildschirmorientiertes, interaktives Programm) ein;
- wählt eine Methode zum Aufbau einer Regel aus,
- inspiziert die graphisch dargestellte Regel und korrigiert sie gegebenenfalls;
- schliesslich aktiviert er den sog. Adviser (Ratgeber) des Programms, der dem Nutzer die Abarbeitung des Programms ermöglicht, indem er ihm Fragen stellt. Nachdem der Adviser genug Informationen für seine Entscheidung erhalten hat, teilt er diese dem Benutzer (oftmals in Gestalt einer Empfehlung) mit. In den Beispielprogrammen von 1st Class wird ihm der zum Essen passende Wein, die für ein bestimmtes Wetter angemessene Jacke empfohlen, er erhält Steuertips, Hinweise auf sein Infarktrisiko bei bestimmten Verhaltensweisen, auf eine eventuelle Disfunktion seiner Nieren, auf die beste Methode eines Gütertransports per Schiff, für die Benutzung von Computern, die Kreditnahme, ein optimales Marketing, die Einschätzung und Auswahl geeigneten Personals usw. First Class akzeptiert als Eingabe auch graphisch gestaltete Regelbäume, es kann mehrere Wissensbasen zusammenfügen, andere in ASCII-Files geschriebene Programme aufrufen, in andere Softwaresysteme integriert werden, und es nutzt statistische Mittel (Wahrscheinlichkeiten), wenn zwischen Alternativen unterschieden werden soll. Ein Modul kann 32 Faktoren, 32 Ergebnisse und 255 Beispiele enthalten.

Weitere bekannte Expertensysteme für IBM-kompatible Rechner sind der «VP-Expert» (WANG/MUELLER, 1989; SAWYER, 1987), ESTA (MALMGREN-HANSEN, 1988) und diverse Programme von «Thinking Software, Inc.» (1987) namens TURBO EXPERT TOOLKIT, ESIE, INFERENCE, EXPERT, dANIMAL, XXXPERT, RULEMAKER und IMPSHELL.

Auch für Macintosh-Computer liegen Expertensysteme bzw. Shells in grosser Zahl vor, etwa die Dortmunder Systeme zur chemischen Prozesssynthese (WELKER, 1989, 111) mit Bezeichnungen, wie «Äzeopert» (Azeotropic Separation Expert System), «Heatpert» (Heat Integration Expert System) oder die Expertensystem-Entwicklungsumgebung «Babylon» (1989, 113) aus Sankt Augustin, die ein hybrides Werkzeugsystem für die Implementierung und den Betrieb von Expertensystemen darstellt und für

den Einsatz an Hochschulen in Vorlesung, Seminar und Praktikum vorgesehen ist. Sie benötigt allerdings bereits mindestens 4–8 MB Hauptspeicher.

Weshalb Expertensysteme heutzutage ein dringendes praktisches Erfordernis sind, wird aus einer Beschreibung von biomedizinischen Expertensystemen deutlich, die an der Universität Bielefeld entwickelt werden: «Bei den medizinischen Expertensystemen geht es um wissensbasierte Computerverfahren, die den Arzt bei komplexen Diagnose- und Therapieentscheidungen unterstützen... Die Kluft zwischen dem Stand medizinischer Forschungserkenntnisse und dem Umfang ihrer Anwendung beim Einsatz in der Patientenversorgung gründet sich hauptsächlich auf die Tatsache, dass praktizierende Ärzte häufig nicht über die jüngsten Daten der Forschung verfügen und zudem Entscheidungen unter beschränkten zeitlichen Ressourcen treffen müssen. Eine Chance für die Anwendungsmöglichkeiten der Künstlichen Intelligenz läge darin, die Expertise und den Informationsstand von Ärzten durch künstliche Systeme zu ergänzen.» (Bielefeld 1990, 1)

All diese Expertensysteme sind, wie deutlich geworden sein dürfte, nicht im Hinblick auf ihre Nutzung für das Sprachenlernen geschrieben bzw. untersucht worden; sie zielen fast ausschliesslich auf technische/technologische, ökonomische, naturwissenschaftliche Bereiche oder auf die Medizin und den Alltagsbereich. Gerade beim Erwerb sprachlichen und landeskundlichen Wissens und Könnens können sie jedoch unserer Auffassung hilfreich sein. Besonders erfolgversprechend scheint uns ihre Einbindung in Hypertextprogramme bzw. ihre Fusion mit Hypertext, wie dies bei KnowledgePro (Thompson, 1987) versucht wird.

Die Designer dieses Systems verfolgen die Absicht, die Schwächen typischer Expertensysteme zu vermeiden. Diese bestehen vor allem in der rigiden, eingeschränkt kommunikativen Frage–Antwort-Struktur. Die feste Vorstrukturierung verhindert die volle Kontrolle des Anwenders über das System. So kann er beispielsweise keine zusätzlichen Informationen zur Beantwortung einer Frage, Worterklärungen usw. erhalten.

Nutzt man bei der Erarbeitung des Systems Hypertext, sind derartige Hilfen per Aktivierung der bereits erwähnten Buttons zu gewährleisten. Wenn diese Buttons mit der zentralen Topicstruktur verknüpft werden, kann der Benutzer den Programmablauf wesentlich mitgestalten. Die Kombination beider Techniken führt zu einer Angleichung des Mensch–Maschine-Dialogs an die natürliche Kommunikation, die je nach individueller Disposition von Anwender zu Anwender unterschiedlich verläuft.

KnowledgePro macht es dem Fachmann relativ leicht, eine Version des Systems auf seinem speziellen Gebiet zu schaffen, d.h. seine eigene Wissensbasis einzugeben und damit verbundene Probleme lösen zu lassen. Er

braucht über keine wesentliche Programmiererfahrung zu verfügen, sondern kann bereits nach der Beherrschung von sechs einfachen Kommandos mit der Erstellung seiner eigenen Version beginnen:

- say	bedeutet: eine Mitteilung machen,
- ask	eine Frage stellen,
- if ... then	eine Entscheidung treffen,
- #m	ein Hypertext-Konzept markieren,
- window	ein Fenster öffnen,
- close window	ein Fenster schliessen.

Das System wird auf zwei 5¼-Zoll-Disketten mit je 360 K geliefert und läuft auf IBM PC, XT, AT und PS2-kompatiblen Rechnern mit 640 K RAM und einer Festplatte.

Schliesslich halten wir es für empfehlenswert, die Expertensysteme für die Weiterentwicklung von Autorenprogrammen zu nutzen; denn sie sind in der Lage, freiere Antworten mit freieren Evaluationsverfahren zu analysieren und auf der Grundlage der ihnen inhärenten Regelsysteme mit komplexen Verzweigungs- und Beratungsstrategien zu reagieren (vgl. SCHULMEISTER, 1989, 46).

Man sollte sich dabei nicht von den meist schwachen Demonstrationsprogrammen, die den Expertensystemen oftmals angefügt sind, täuschen lassen. Ähnlich wie bei vielen Autorenprogrammen und -systemen wurden sie offenbar von den die Expertensysteme verfassenden EDV-Spezialisten, die weder Fachleute auf dem jeweiligen ökonomischen, medizinischen etc. Gebiet noch ausgebildete Pädagogen sind, gleich mitgeliefert, so daß ein deutlicher Bruch zwischen der meist unzweifelhaften Professionalität im programmiertechnischen Bereich und einem teilweise peinlichen Dilettantismus in den Anwendungsbereichen zustande kam. Letzterer wird auch häufig bei der Gestaltung der Bedienungsoberfläche deutlich, die gegen die Gesetze der Ergonomie verstösst. Infolgedessen wäre die Schaffung eigener Expertensysteme in Zusammenarbeit von Informatikern, Linguisten/Landeskundlern, Pädagogen und Psychologen unter Nutzung der bisher angebotenen Varianten natürlich der beste Weg für die künftige Einbeziehung dieser intelligenten Nebenprodukte der Forschung zur Künstlichen Intelligenz in das Sprachenlehren und -lernen. Diese könnten, wie gesagt, in der Verbindung mit Hypertext und Parsern die Produktion einer neuen Generation von Spracherwerbsprogrammen wesentlich mitbestimmen, die dem natürlichen Lernprozess besser entsprechen als die Masse der gegenwärtig angebotenen Computerprogramme und den Lerner somit effektiver auf die sprachlichen Praxisanforderungen, auf natürliches Sprachhandeln vorbereiten, damit es dem Sprecher nicht wie jenem Studenten mit seinem

Verstand geht, von dem Lichtenberg schreibt: «Wenn er seinen Verstand gebrauchen sollte, so war es ihm, als wenn jemand, der beständig seine rechte Hand gebraucht hat, etwas mit der linken tun soll.»

Friedrich-Schiller-Universität Jena
Institut für Auslandsgermanistik
DDR Jena

GERHARD WAZEL

Literatur

ASHWORTH, David/STELOVSKY, Jan (1989): «Kanji City. An Exploration of Hypermedia Applications for CALL.» In: *Calico Journal 4,* 27–39.

BABYLON, FORSCHUNGSGRUPPE EXPERTENSYSTEME (1989). In: *Apple Computer GmbH,* München, 113–114.

BEGEMAN, Michael/CONKLIN, Jeff (1988): «The Right Tool for the Job.» In: *Byte, October,* Hightstown, McGraw-Hill, 255–266.

BIELEFELDER UNIVERSITÄTSZEITUNG (1990): Nr. 156 (ohne Autorenangabe), 1–2.

BIRKHAN, Georg (1989): «Bedarf an und Nachfrage nach Computerwissen für den Hochschulunterricht in den Sozialwissenschaften.» In: SCHULMEISTER, Rolf (Hrsg.): *Computereinsatz im Hochschulunterricht.* Ammersbeck bei Hamburg, Lottbeck Jensen, 221 p., ISBN 3-926987-12-X.

BLATT, Achim/FREIGANG, Karl-Heinz/SCHMITZ, Klaus-Dirk/THOME, Gisela (1985): *Computer und Übersetzen,* Hildesheim/Zürich/New York, Georg Olms, 331 p.

BOURNE, John et al. (1989): «Intelligent Hypertutoring in Enginneering.» In: *Academic Computing,* Sept., McKinney, TX, 18–20, 36–48.

BURSTON, Jack (1989): «Towards Better Tutorial CALL: A MATTER of Intelligent Control.» In: *CALICO JOURNAL 4,* Provo, 75–89.

BURTNESS, Larry (1989): «A Look at IBM-Linkway.» In: *Hyperlink Magazine 3,* Atlanta, 36–30.

BURTNESS, Larry (1989a): «A Videodisc Browser.» In: *Hyperlink Magazine 4,* Atlanta, 11–47.

CHIGNELL, Mark/LACY, Richard (1988): «Project Jefferson: Integrating Research and Instruction.» In: *Academic Computing, October,* McKinney, TX, 12–17, 40–45.

COOK, Vivian (1988): «Designing a BASIC Parser for CALL.» In: *CALICO JOURNAL 1,* Provo, 50–67.

FIDERIO, Janet (1988): «A Grand Vision.» In: *Byte, October,* Hightstown, McGraw-Hill, 237–246.

FRISSE, Mark (1988): «From Text to Hypertext.» In: *Byte, October,* Hightstown, McGraw-Hill, 247–253.

FROMMER, Judith (1989): «Listening, Looking, and Learning with Mac Lang. In: *CALICO JOURNAL 4,* Provo, 51–71.

GRIESSHABER, Wilhelm (1989): «Computereinsatz im Hochschulunterricht in den Sprachwissenschaften.» In: SCHULMEISTER, Rolf (Hrsg.): *Computereinsatz im Hochschulunterricht.* Ammersbeck beim Hamburg, Lottbeck Jensen, 221 p., ISBN 3-926987-12-7.

GUIDE/GUIDE 2 (1988): *Setup Manual* (ohne Autorenangabe), Bellevue, OWL International, 165 bzw. 23 p.

HANDKE, Jürgen (1989): «Computer Assisted Language Learning und Künstliche Intelligenz.» In: *Die Neueren Sprachen 88,* 1, 21–32.

HAPGOOD, William (1986): *First class,* Wayland, Programs in Motion Inc., Disketten.

HARVEY, Greg (1988): *Understanding HyperCard.* San Francisco/Paris/Düsseldorf/London, Sybex, 581 p., ISBN 0-89588-506-9.

HELLWIG, Harold (1987): «Problems and Solutions in Parsing.» In: *CALICO JOURNAL 1,* Provo 49–64.

IBM LinkWay (1989): FM (ohne Autorenangabe), Boka Raton, IBM Corp., ca. 200 p.

JAFFE, Conrade/LYNCH, Patrick (1989): «Hypermedia for Eduction in the Life Sciences.» In: *Academic Computing, Sept.,* McKinney, TX, 10–13, 52–57.

JUNG, Udo (1989): Rezension zu Leonard, J. «Computers in Language and Literary Work». In: *Die Neueren Sprachen 88,* 1, 119–120.

KINDERMANN, Jörg (1987): *Experten parsing: parsing und Wissenspräsentation,* Hamburg, Helmut Buske Verlag, 53 p., ISBN 3-87-118-838-7.

KRASHEN, Stephen (1985): «The Power of Reading.» Plenary address at the 19th annual TESOL convention. New York.

LEGENHAUSEN, Lienhard/WOFF, Dieter (1989): «Lernerstrategien bei der Textrekonstruktion: STORYBOARD als Übung im Fremdsprachenunterricht.» In: *Die Neueren Sprachen 88,* 1, 3–20.

LORITZ, D. (1987): «An Introductory Lisp Parser.» In: *CALICO JOURNAL 1,* Provo 51.

MALMGREN-HANSEN, L. (1988): ESTA, Brondby, Prolog Development Center, Aps., Diskette.

MANDL, H./HUBER, G. L. (Hrsg.) (1978): *Kognitive Komplexität.* Hogrefe.

MANDL, H./FISCHER, P. M. (Hrsg.) (1985): *Lernen im Dialog mit dem Computer.* München/Wien/Baltimore.

MEYER, Pawel (1989): «Hochgestapelt. HyperCard: MacIntoshs Alleskönner.» In: *c't, Mai,* 88–94.

MULTHAUP, Uwe (1989): «Wie nützlich sind Autorensysteme? Einsatzmöglichkeiten des Computers im Fremdsprachenunterricht.» In: *Praxis des neusprachlichen Unterrichts 2.* Dortmund, Lambert Lensing, 122–130.

MURRAY, Janet/MORGENSTERN, Douglas/FURSTENBERG, Gilberte (1989): «The Athena Language-Learning Project: Design Issues for the Next Generation of Computer-Based Language-Learning Tools.» In: SMITH, Flint (Hrsg.): *Modern Technology in Foreign Language Education: Applications and Projects.* Lincolnwood, National Textbook Company, 368 p., ISBN 0-8442-9387-3.

NAKAJIMA, Kazuko (1988): «On Developing HyperCard Stacks for the Study of Chinese Characters: KanjiCard.» In: *CALICO JOURNAL 2,* Provo, 75–87.

OTTO, Frank (1989): «Special CALICO Programs and Activities.» In: *CALICO JOURNAL 4,* Provo, 3–5.

RAGHAVAN, Kalyani (1989): «Smithtown: An Intelligent Tutoring System.» In: *T. H. E Journal 1,* Tustin, 50–53.

RÜSCHOFF, Bernd (1989): «Selbständiges Lernen mit dem Computer am Beispiel interaktiver audiovisueller CALL Materialien.» In: *Die Neueren Sprachen 88,* 1, 50–60.

SANDERS, Alton/SANDERS, Ruth (1987): «Designing and Implementing a Syntactic Parser.» In: *CALICO JOURNAL 1,* Provo, 77–86.

SANDERS, Alton/SANDERS, Ruth (1989): «Syntactic Parsing: A Survey». In: *Computers and the Humanities 23,* Kluwer Academic Publishers, Paradigm Press, 13–30.

SAWYER, B.: *VP-EXPERT,* Berkeley, Idea Ware Inc. 1987, Disketten.

SCHULMEISTER, Rolf (1989): *Computereinsatz im Hochschulunterricht.* Ammersbeck bei Hamburg, Lottbeck Jensen, 221 p., ISBN 3-926-987-12-7.

SMITH, Flint (Hrsg.) (1989): *Modern Technology in Foreign Language Education. Applications and Projects,* Lincolnwood, National Textbook Company, 768 p., ISBN 0-8442-9387-3.

SPADA, H./OPWIS, K. (1985): «Intelligente tutorielle Systeme aus psychologischer Sicht.» In: MANDL, H./FISCHER, P. M. (Hrsg.): *Lernen im Dialog mit dem Computer.* München/Wien/Baltimore.

SZABO, Janos/SZASZ, Ferenc (Hrsg.) (1989): *Theorien, Epochen, Kontakte. Teil 2,* Budapest, Loràn-Eötvös-Universität, 235 p., HU ISSN 013-905-X.

TAZELAAR, Jane (1988): «In Depth Hypertext.» In: *Byte, October,* Hightstown, McGraw-Hill, 234.

THINKING SOFTWARE, INC., 46–16 65th Place, Woodside, New York, 11377, Disketten.
THOMPSON, Bev/Bill (1988): Knowledge Pro, Nassan, Knowdledge Garden, Inc., ca. 60 p.
UNDERWOOD, John (1989): «HyperCard and Interactive Video.» In: *CALICO JOURNAL 3,* Provo, 7–20.
UNDERWOOD, John (1989a): «Hypermedia: Where We Are and Where We Aren't.» In: *CALICO JOURNAL 4,* Provo, 23–26.
WANG, Wally/MUELLER, John (1989): *Illustrated VP-Expert,* Plano, Wordware Publishing, 241 p., ISBN 1-556-22057-X.
WAZEL, Gerhard (Hrsg.) (1987): *Computer und Video im fremdsprachlichen Deutschunterricht.* Jena, Wissenschaftliche Beiträge der Friedrich-Schiller-Universität, 242 p., ISSN 0232-3753.
WAZEL, Gerhard (1988): «Funktion und Struktur von Autorenprogrammen für den computergestützten Fremdsprachenunterricht.» In« *Germanistische Mitteilungen 28,* Bonn, F. Dümmler Verlag, 25–37.
WAZEL, Gerhard (1990): «Lieber weniger, aber besser. Versuch einer Zwischenbilanz zum Computereinsatz im Fremdsprachenunterricht.» In: *Deutsch als Fremdsprache 1,* Leipzig, Enzyklopädie, 2–8.
WELKER, R. (1989): «Expertensysteme in der Technischen Chemie.» In: *Apple Computer in Forschung und Lehre.* München, Apple Computer GmbH, 111.
WEST, Gaby (1989): «An Interactive Video Program.: TOBIC.» In: *CALICO JOURNAL 3,* Provo, 51–59.
YOUNG, Richard (1988): «Computer-Assisted Language Learning Conversations: Negotiating an Outcome.» In: *CALICO JOURNAL 3,* Provo, 65–83.