

## **Neue Medien für den Russischunterricht (Computergestütztes Lehren und Lernen mit HyperCard)**

### **0. Vorrede**

Der vorliegende Artikel soll zur Diskussion eines Themas in der Slavistik anregen, das sich zwischen Computerlinguistik, dem sog. Information Retrieval, Angewandter Linguistik und der Didaktik des Russischen bewegt<sup>1</sup>, wobei das Interesse hier insbesondere den Perspektiven gilt, die eine Nutzung qualitativ neuer Medien für Unterricht wie Studium eröffnen würde, nicht so sehr neuen Technologien als solchen. Selbst wenn – gerade in den Geisteswissenschaften, und hier besonders in der Slawistik, weniger in Fächern wie der Anglistik – vieles noch Utopie ist und eine undistanzierte Euphorie verfehlt wäre, darf dies kein Grund dafür sein, die Möglichkeiten und Perspektiven, die sich im Bereich des Computereinsatzes bieten, nicht wenigstens zur Kenntnis zu nehmen und kritisch zu begleiten.

Der Einsatz von Personal Computern prägt die gesamte Arbeitsweise unserer Gesellschaft von Tag zu Tag in stärkerem und ganz augenfälligem Maße. So verwundert es nicht, daß auch bei den mit schöner Regelmäßigkeit aufflammenden Diskussionen über eine Neugestaltung der Studieninhalte bzw. über zusätzliche Qualifikationen, die die Chancen der Absolventen auf dem Arbeitsmarkt erhöhen, das Vertrautsein mit Personal Computern immer wieder ins Spiel gebracht wird. Wenn es um den Einsatz von PCs im Unterricht geht, sollte man deshalb ruhig auch diesen Nebeneffekt im Auge behalten. Ziel des vorliegenden Beitrages kann es jedoch nicht sein, das Für und Wider des Computereinsatzes zu behandeln; es soll hier einzig um die Möglichkeiten gehen, die mit dem Begriff 'Hypertext' (s.u.) verknüpft sind, und darum, einige Einsatzmöglichkeiten solcher Systeme im slawistischen Hochschulunterricht aufzuzeigen. Da Hypertext als Konzeption einer breiteren – auch wissenschaftlichen – Öffentlichkeit noch nicht geläufig ist, soll vor den speziell slawistischen Aspekten der Kontext kurz hergestellt werden, in dem diese Neuerungen stehen.

### **1. HyperText, HyperMedia**

Im Laufe des Jahres 1987 gewann das HyperText-Konzept aus einer Reihe von Gründen über den engen Zirkel der an HyperText-Systemen arbeitenden Computer-Spezialisten hinaus Bedeutung und wurde auch einer breiteren Öffentlichkeit bekannt, was sich z.B.

---

<sup>1</sup> Z.T. wird das modische Kürzel CALL, für Computer Aided Language Learning, benutzt, das ganz offensichtlich nach dem Vorbild der schon geläufigeren CA-Abkürzungen (CAD, CAM) geprägt ist.

daran zeigt, daß eine Flut von Artikeln zu diesem Thema in den verschiedensten Publikationen erschien, eine Konferenz zu diesem Thema abgehalten wurde (Hypertext '87) und renommierte Computerzeitschriften wie z.B. die 'Communications of the acm' HyperText eine Sondernummer widmeten (1987/7), ähnlich CD-ROM Review (1988/3), andere die Titelgeschichte (BYTE 10/1988, online 1988/2), Bibliographien zusammengestellt wurden (FRANKLIN 1988) und dergleichen mehr.

Das Ereignis, das das HyperText-Konzept einer breiteren Öffentlichkeit interessant machte, war ohne Zweifel die Veröffentlichung des HyperCard-Programmes für den Apple Macintosh im Sommer 1987, das, wie der Name schon andeutet, HyperText-Vorstellungen in die (elektronische) Wirklichkeit umsetzt. Vor allem die Tatsache, daß dieses Programmpaket jedem neuen Gerät kostenlos beigelegt wird, sorgte praktisch augenblicklich für eine weite und allgemeine Verbreitung. Dieses Programm wurde zu Recht als so revolutionär empfunden, daß es eine für ein Computerprogramm außergewöhnlich breite Resonanz in allen möglichen Medien, auch außerhalb von Fachpublikationen, fand.

An einem Beispiel läßt sich am einfachsten verdeutlichen, was unter HyperText zu verstehen ist. Jedermann kennt die klassischen Bibliothekskataloge, die auf dem Prinzip der linear angeordneten Information aufbauen. Jeder Autor und jedes Buch steht hier in streng alphabetischer Reihenfolge auf einer Karteikarte in einem Karteikasten. Wenn man bei einem Autor ist und beispielsweise einen Querverweis auf Bücher findet, die von diesem Autor herausgegeben wurden, aber unter dem Titel des Werkes, nicht unter dem Namen des Herausgebers, eingeordnet sind, so muß man, will man die Informationen zum zweiten Werk einsehen, den gerade geöffneten Kasten schließen, am Regal den suchen, der das betreffende Werk enthält, ihn öffnen, um dann die betreffende Karte zu suchen. Wie jeder weiß, viel Aufwand für einen eigentlich geringen Ertrag. Dieser manuelle Aufwand wird durch die Verwendung von Microfiches in ‚modernen‘ Bibliotheken zwar geringer, das Verfahren als solches, vor allem aber die sequentielle Anordnung der vorhandenen Informationen, bleibt aber gleich.

Wieviel praktischer wäre es, wenn man sofort und ohne Aufwand und Umweg von jeder Karte zu einer beliebigen anderen Karte blättern – und auch wieder zur ersten Karte zurückkehren – könnte. Genau dies ist der Grundgedanke eines Hypertext-Systems, nämlich die direkte, nichtsequentielle, elektronisch realisierte Verknüpfung von Textpassagen bzw. ganz allgemein von Informationen, die es erlaubt, von einer Stelle zu einer anderen zu springen, wenn dies gewünscht wird, ohne daß dabei die lineare Anordnung der Information eine Rolle spielt. Hypertext-Systeme sind im weitesten Sinne also Datenbanken, in denen Texte oder andere Informationen gespeichert sind.<sup>2</sup> Im Endzustand könnte das

---

<sup>2</sup> Von „normalen“ Datenbanken unterscheidet sich HyperText dabei auch durch die Terminologie: während man Informationen traditionell in *Datensätze* (Records) und jeden Datensatz in *Felder* (Fields) gliedert, wobei in den sogenannten relationalen Datenbanken Felder verschiedener *Dateien* (Files) miteinander verknüpft werden können, spricht man in HyperText-Systemen von *Knoten* (Nodes), die miteinander (hierarchisch oder nichthierarchisch) verknüpft werden, wobei jeder solcher Knoten aus einem Bildschirm voll Information besteht. Das hier benutzte Bild von Knotenpunkten, die in einem dreidimensionalen Raum miteinander durch beliebig viele Bahnen verknüpft sind, entspricht dem erklärten Ziel, den Aufbau des menschlichen Gehirns zu simulieren (vgl. auch unten).

gesamte Wissen, das der Menschheit zur Verfügung steht, auf diese Weise bereitgehalten und zugänglich gemacht werden – so jedenfalls die Vision, die wir als solche hier unkommentiert lassen wollen.

Es ist schon modisch geworden zu sagen, daß wir in einem Zeitalter der Informationsflut, der Überschwemmung mit Information leben: unglaublich viel Information ist an allen möglichen Stellen, in Bibliotheken, in Datenbanken, in Archiven etc. vorhanden, aber es besteht oft keine vernünftige Möglichkeit, aus diesem Wust von Informationen die gewünschten herauszusuchen, weil jedes System nach seinen eigenen Spielregeln arbeitet, die erst einmal erlernt werden wollen, der Zugang zu schwierig oder für Außenstehende unmöglich ist etc. Was nützt es z.B., wenn man weiß, daß bei der GMD (Gesellschaft für Mathematik und Datenverarbeitung) in Bonn das „Grammatische Wörterbuch“ von ZALIZNJAK (1977) komplett als Datenbank vorhanden ist, aber nur inakzeptable (kommerzialisierte) Auskunfts- und Zugangsmöglichkeiten bestehen? In Verbindung mit den sich entwickelnden einheitlichen Benutzeroberflächen der Personal Computer, neuen Speichermedien wie den CD-ROM-Laufwerken, die gigantische Datenmengen bereithalten können, wäre Hypertext eine Möglichkeit, diesen Informationsreichtum einer breiteren Nutzung zuzuführen.

Ein anderes Beispiel: jeder weiß, wie umständlich es für einen Leser ist, wenn Fußnoten ans Ende eines Kapitels oder Buches gesetzt werden. Denn was muß der Leser tun: er blättert beispielsweise vom Haupttext zur betreffenden Anmerkung, um zu entdecken, daß auf weiterführende Literatur verwiesen wird, woraufhin er dann im Literaturverzeichnis die eigentliche Information nachschlagen kann, um danach zur Textpassage zurückzukehren, die ihn ursprünglich zu dieser Blätterei veranlaßte. In einem HyperText-System würden Anmerkungen und Literaturangaben sofort sichtbar, wenn man die betreffende Textpassage liest, ein manuelles Suchen entfiel also. In den traditionellen Druckerzeugnissen sind im Grunde alle Randbemerkungen, also das, was – vom Autor oder vom Leser – wirklich auf den – normalerweise unbedruckten – Rand geschrieben wird, Versuche, HyperText-Strukturen in einen linearen Text zu bringen.

Oder man denke, viertes Beispiel, an empirische Untersuchungen, die oft in der einen oder anderen Weise Zahlenmaterial in Form von Tabellen präsentieren, die, damit sie den eigentlichen Fluß der Darlegungen nicht stören, am Ende einer Untersuchung als Anhang angebracht werden.<sup>3</sup> Eine Übertragung in ein HyperText-System könnte nicht nur die Tabelle(n) sofort sichtbar werden lassen, wenn auf sie referiert wird, sondern z.B. auch die Werte, auf die sich die Diskussion im Text gerade bezieht, optisch hervorheben.

Hypertext-Systeme konkurrieren also in starkem Maße mit der überkommenen Form der Wissensvermittlung in Gestalt von Büchern, und sie suchen ihre Rechtfertigung in den nicht zu leugnenden Mängeln dieses Mediums. Auf eine kurze Formel gebracht, lautet das wichtigste Argument für Hypertext-Systeme: “You can’t ever find anything in a book when you want to”, während eine ebenso pointierte Verteidigung lautet: “You can’t

---

<sup>3</sup> Vgl. als Beispiel für diese Art des Buchlayouts KRYSIN (1974). Im traditionellen Print-Medium ist eine solche Anordnung des Materials oft tatsächlich der einzig gangbare Weg, wenn man den eigentlichen Text nicht vollkommen auseinanderreißen will.

get lost in a book”. Das Pro-Argument stützt sich auf die Erfahrung, daß man sich oft erinnert, bestimmte Dinge in einem Buch gelesen zu haben, die üblichen Indizes aber selten genug in der Lage sind, den Ort zu benennen, wo das assoziativ Erinnernte zu finden ist. Das Contra-Argument richtet sich demgegenüber gegen die potentiell unendlichen Verknüpfungen, die in globalen Hypertext-Systemen möglich werden und gegen die Möglichkeit, von Information zu Information zu springen und dabei vollkommen den Überblick über die Struktur des dargebotenen Wissens zu verlieren (bzw. ihn überhaupt nicht erst zu gewinnen). Auch aus pädagogischen Gründen befürwortet nicht jeder solche weitverzweigten Verknüpfungen, vgl. z.B. LEEMON (1988, 49), der die Gefahr zu großer Ablenkung befürchtet:

“...do we really want to give hypertext to young school children, who already have plenty of distractions? After all, if a child is studying a lesson in ancient history, we really don’t want him to click on the section where the Chinese invent gunpowder and end up in a chemistry lesson on how to create fireworks in the basement. Such a student might be better served by a HyperCard application, one in which every distraction which he chooses to avoid studying leads him right back to the subject he’ll be tested on.”

Ich glaube, daß beide Argumente keine ernsthaften Einwände gegen den Einsatz eines HyperText-Systems in dem Bereich sein können, um den es im vorliegenden Beitrag geht, um den Unterricht bzw. das Selbststudium nämlich. Bei der Einführung solcher Systeme geht es zunächst ganz einfach darum, daß der Lehrende bei seiner Vorbereitung oder auch im Unterricht schnellen Zugriff auf alle möglichen Informationen, die gerade benötigt werden, hat, bzw. darum, Übungsaufgaben für Studenten in einer Weise zu gestalten, die die Nachteile des Schriftmediums vermeidet. Auch gibt es systemintern genug Möglichkeiten, wie derjenige, der solche Übungen zusammenstellt, die genannten – in diesem Zusammenhang vielleicht tatsächlich unerwünschten – Ablenkungseffekte verhindern kann (forschende Neugier an sich ist ja nichts Verwerfliches).

Um an dieser Stelle gleich einem möglichen Mißverständnis vorzubeugen, sei folgender Punkt betont: es kann beim Einsatz von Computern mit Hypertext-Software nicht darum gehen, die traditionelle Form der Wissensvermittlung durch das Buch *abzulösen*. Vielmehr geht es darum, sich der Möglichkeiten und Vorteile beider Prinzipien und Arbeitsweisen für die verschiedenen Anwendungszwecke zu bedienen.

Als Beispiel, wie eine solche Verknüpfung von Bild und Ton aussehen kann, sei auf eine mit HyperCard realisierte Demonstration dieser Techniken verwiesen. Diese Datei mit dem Titel “Bill Sez...”, deren Inhalt, soweit er im Druck wiedergegeben bzw. umgesetzt werden kann, in **Abb. 1** zu finden ist, beinhaltet als Grundlage eine Aufnahme mit einem ganz gewöhnlichen Kassettenrecorder. Zu hören ist ein Ausschnitt aus einem Diskussionsbeitrag des HyperCard-Programmierers Bill Atkinson aus dem Jahre 1984, der seine Visionen von dem beschreibt, was er dann einige Jahre später als fertiges Produkt vorgelegt hat. Zu diesem Text, der hier – in der Qualität einem Kassettenrecorder vergleichbar – zu hören ist, werden gleichzeitig Bilder und Bildfolgen gezeigt, die sich jeweils auf das gerade Gesagte beziehen und es in eine neue Dimension heben.


**Bill Atkinson**

**“The Vision of HyperCard”**


*recorded at Hackers 1984 by Fabrice Florin*


*Sound Stack produced by Tim Oren*

Click the mouse for a talk about the origins of HyperCard...





*“Imagine using a computer to help you...  
People ask why do lightbulbs burn out? And you see a  
picture of a light bulb and other notes of information  
and you can just touch on and, say, browse and steer  
through a large amount of information in a pleasant and  
enjoyable way.  
I mean, can you imagine a person now reading an  
encyclopedia cover to cover? Doesn't happen.  
But people spend hours and hours playing adventure,  
exploring some imaginary information.  
Well, if it were pleasurable and enjoyable to explore  
real live all human knowledge, uhm, suddenly you got a  
powerful learning tool.  
Well I can't afford to take the time to find out why light  
bulbs burn out. I still don't know.”*





**HyperCard!**





**Bill Atkinson**

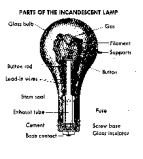

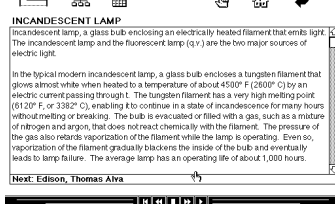

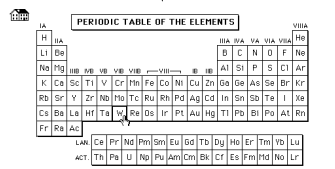

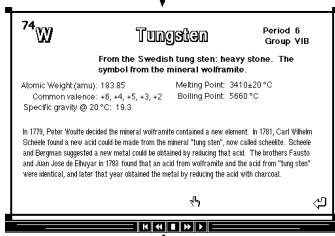

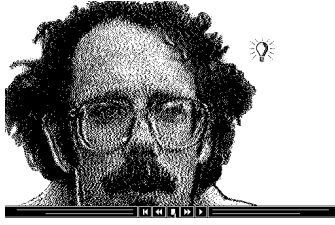










Abb. 1: HyperCard-Stapel “Bill Sez...”

Wenn Atkinson als Beispiel das Durchbrennen einer Glühbirne anspricht, dann zeigt die visuelle Ebene eine nichtlineare Verknüpfung von Informationen, wie sie in HyperText-Systemen möglich ist: vom eingangs gezeigten Bild des Redners geht es zur Darstellung einer Glühbirne mit ihren Bestandteilen, dann weiter zu einer Tabelle chemischer Elemente und zu einer biographischen Notiz über die Person Tungsten, danach wieder zurück zum Autor selbst. Die Verknüpfung dieser Bilder miteinander ist in **Abb. 1** ebenfalls ersichtlich, wenn auch der Gesamteffekt schriftlich schlecht beschrieben und vom Leser sicher noch schlechter nachvollzogen werden kann.

An dieser Stelle mag es sinnvoll sein, zunächst einen Blick zurückzuwerfen auf Ursprünge und Herkunft der Hypertext-Ideen.

Die Geschichte der Hypertext-Konzeption reicht in die 40er Jahre zurück. Damals beschrieb V. BUSH (1945) ein “memex” (für memory expander) genanntes “personal information management system”, das, wie der menschliche Verstand, auf assoziativer Verknüpfung von Information aufbauen sollte, wobei er damals annahm, diese Informationen wären auf Microfilmen gespeichert. Als Gerät realisiert wurde ein solches System nicht, aber die Grundgedanken waren formuliert.

Weiterentwickelt wurden diese Vorstellungen seit den sechziger Jahren von D. ENGELBART am Stanford Research Institute, dessen Arbeiten zu einem kommerziellen System (“Augment”) geführt haben (1963, 1968), und von T. NELSON mit seinem “Project Xanadu” (das bisher nur in literarischen Visionen und Skizzen existiert, vgl. 1974, 1981). Es sollte in seinem Endausbau die gesamte Weltliteratur zur Verfügung stellen, wobei sogar schon geregelt war, wie das Honorar der elektronisch gelesenen Autoren zu berechnen wäre. Von NELSON stammt auch der Begriff “HyperText” für nichtsequentielles Schreiben.

Die genannten (Vor-)Arbeiten und Überlegungen mündeten in weitere konkrete Projekte ein, von denen die bekanntesten heute das “NoteCard”-Projekt von Xerox PARC (Palo Alto Research Center) und das “Intermedia”-Projekt der Brown University sind.<sup>4</sup>

Es ist – zumindest aus der Sicht des Autors – so bezeichnend wie erfreulich, daß die wichtigsten Projekte an Universitäten (und nicht in der freien Wirtschaft) entwickelt werden, wobei die Universitäten es sich in der ersten Stufe zur Aufgabe gestellt haben, ihren Mitgliedern – Lehrenden und Lernenden – bestimmte Informationen in einem Hypertext-System zur Verfügung zu stellen, um Curricula, z.B. für Grundkurse, zu entwickeln bzw. zu nutzen. Und anders auch als die Forschungen zur künstlichen Intelligenz, die weithin von einem ganz bestimmten militärischen Interesse beherrscht sind, sind die angesprochenen HyperMedia-Projekte davon im wesentlichen frei.

Auch wenn man weiterhin meist von *HyperText*-Systemen spricht, so ist die geschilderte Konzeption im Laufe der Zeit mit den immer größeren Möglichkeiten immer kleinerer Computer zu einem *HyperMedia*-Konzept gewachsen. Verknüpft werden sollen nicht mehr nur Textpassagen, sondern Texte mit dazugehörenden Bildern, Graphiken,

---

<sup>4</sup> Genaueres zu Hintergrund, Problemen und Ausblicken vergleiche bei CONCLIN (1987), in CD-ROM Review (1988), SMITH/WEISS (1988), YOUNG (1986), SMITH (1988), FRANKLIN (1988), FIDERIO (1988), FRISSE (1988), BEGEMAN/CONKLIN (1988).

Tondokumenten, Filmpassagen etc. Auch komplexe Zusammenhänge können bekanntlich durch eine optische Präsentation leichter faßbar und nachvollziehbar gemacht werden: ein Bild, so weiß das Sprichwort, sagt mehr als 1000 Worte, und, so ließe sich jetzt hinzufügen, eine Bildfolge, ein Kurzfilm sozusagen, mehr als Standbilder.<sup>5</sup>

Im übrigen gibt es auch eine interessante – und sicher nicht zufällige – inhaltliche Parallele zur Linguistik: das Hypertext-Konzept beruht, wie schon angedeutet, auf der Idee, eine nichtsequentielle Informationsspeicherung und eine nichtsequentielle Informationsrückgewinnung zu ermöglichen, und es ist ja bekannt, daß die Informationsspeicherung und -verknüpfung im menschlichen Gehirn tatsächlich assoziativ geregelt sind; in der Linguistik nun ist in letzter Zeit eine ganz ähnliche Konzeption unter dem Schlagwort der ‚neuronalen Verknüpfungen‘, die es zu untersuchen gelte, entwickelt worden. Beide Disziplinen versuchen also im Grunde, einen und denselben Aspekt des menschlichen Verstandes nachzubilden, zu modellieren. Über das rein praktische Interesse hinaus, wie Hypertext-Modelle für Lehre und Forschung als Hilfsmittel eingesetzt werden können, gibt es also durchaus auch einen sachlichen Grund, sich für diese Dinge zu interessieren. Selbst künftige Computer sollen auf einer ähnlichen Grundlage arbeiten (‘neuronal computers’).

## 2. HyperCard

Der Anlaß, über die Einsatzmöglichkeiten von HyperText-Medien in der Russistik nachzudenken, war, wie schon deutlich geworden ist, die Verfügbarkeit eines Programmes mit dem Namen HyperCard, das zur Standardausrüstung für den Apple Macintosh gehört.

HyperCard ist ein Produkt, das sich einer einfachen Kategorisierung widersetzt. Am besten läßt es sich vielleicht als ein interaktives Werkzeug zum Informationsmanagement beschreiben, denn diesem Ziel ordnen sich die verschiedenen Komponenten des Programmes unter. Es ist nämlich gleichzeitig Datenbank, Graphikprogramm und Programmiersprache (HyperTalk genannt).<sup>6</sup> Eine der Anwendungen dieses Programmes, an die zunächst gar nicht so sehr gedacht war, die sich heute aber als ein Haupteinsatzbereich durchgesetzt hat, ist sein Einsatz als „Autorensystem“, d.h. als Werkzeug zum Erstellen von Unterrichtssoftware. Auch wenn es – für die verschiedensten PCs – schon seit einigen Jahren solche Autorensysteme gibt, schafft doch erst HyperCard mit der Kombination aller Möglichkeiten, die es bietet, völlig neue Perspektiven in diesem Bereich.

---

<sup>5</sup> Inzwischen spricht man etwas allgemeiner auch von “Multimedia”, einem Einsatzbereich, bei dem HyperCard eine ganz wichtige Rolle als Steuerzentrale spielt, in dieser Funktion mit anderen Programmen aber durchaus konkurrieren muß. Eine ausführliche Vorstellung multimedialer Projekte findet sich in MacUser (1989/3).

<sup>6</sup> Von den vielen Artikeln zu HyperCard seien hier nur einige genannt: THORNBURG (1988a, 1988b), GOODMAN (1987a, 1987b), MacUp (1988), WILLIAMS (1987), CUSTER (1987), WHEELS FOR THE MIND (1987).

HyperCard benutzt die Metapher von Karteikarten, um Informationen, die Text oder Graphik sein können, aufzunehmen und zu präsentieren. Karten haben die Größe eines Macintosh-Standardbildschirmes (und haben damit in etwa die Größe eines Drittels einer A4-Seite); es kann jeweils eine Karte betrachtet werden.<sup>7</sup> Karten fügen sich zu Stapeln zusammen, wobei normalerweise in einem Stapel Karten mit ähnlich strukturierter Information und ähnlichem Aussehen zusammengefaßt werden, was aber keineswegs zwingend ist. Diese ‚Stapel‘ entsprechen dem, was normalerweise eine ‚Datei‘ genannt wird.



Abb. 2: Startkarte (Home Card)

Startet man HyperCard, so gelangt man zur sogenannten ‚Home Card‘ (deutsch: ‚Startkarte‘, vgl. **Abb. 2**).<sup>8</sup> Auf dieser Karte sind die Stapel eingetragen, die mit dem Programm mitgeliefert werden; die Einträge können vom Nutzer seinen eigenen Bedürfnissen beliebig angepaßt werden. Auf diese Stapel kann man zugreifen, indem man einfach mit der ‚Maus‘ (so heißt das Eingabegerät, das man mit der Hand auf einer Unterlage bewegt, um analog einen Zeiger auf dem Bildschirm zu steuern) auf das entsprechende Bild zeigt und einmal die Taste dieses Eingabegerätes (kurz: ‚Maustaste‘) betätigt. Das ist für das Programm das Signal, die Startkarte zu verlassen und die gewünschte Datei zu öffnen.

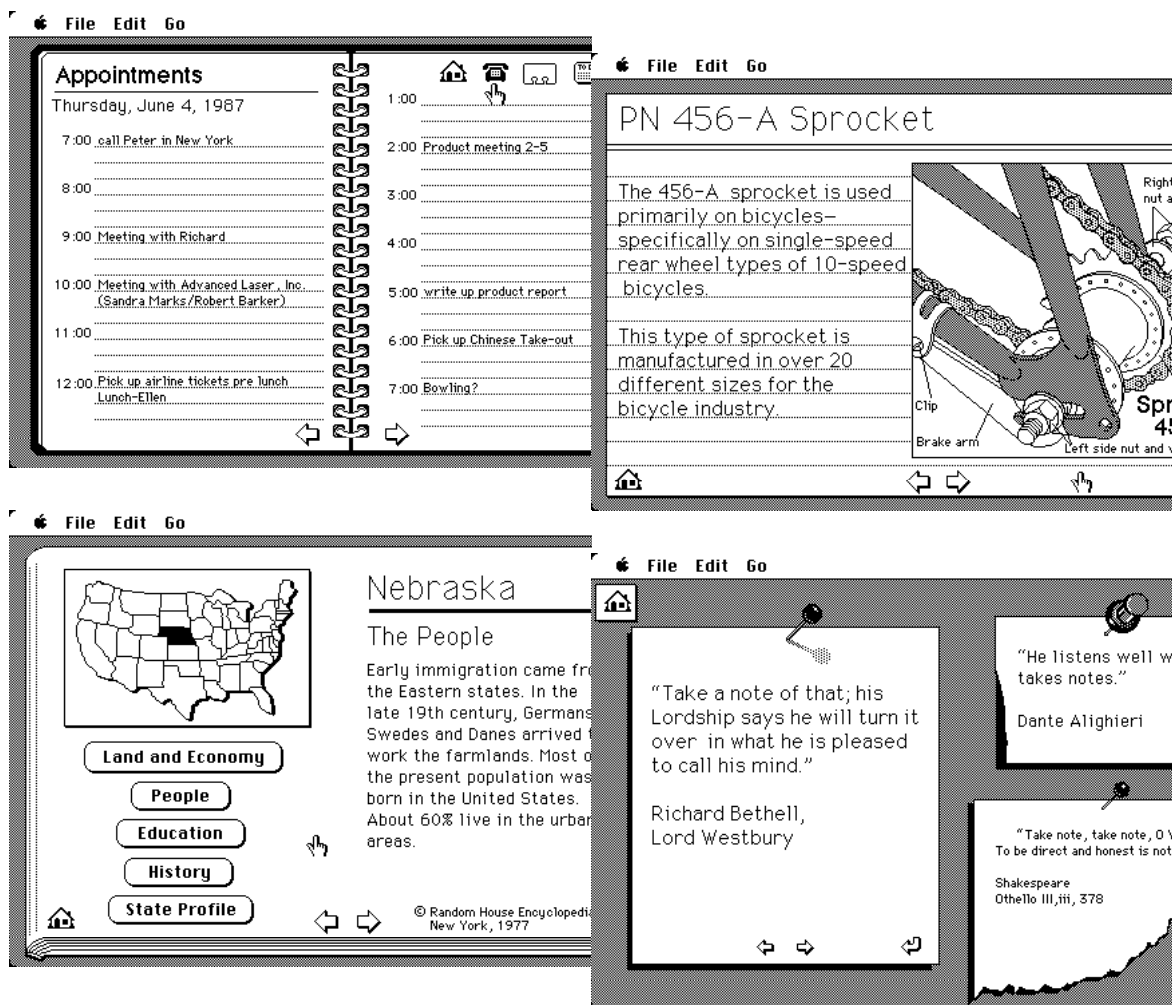
<sup>7</sup> Die Beschränkung auf die Bildschirmgröße und auf jeweils eine Karte sind die zwei wichtigsten Kritikpunkte, die gegen die erste Version von HyperCard vorgebracht wurde; es ist damit zu rechnen, daß dieses Handicaps in einer künftigen Version wegfallen werden.

<sup>8</sup> Diese und die folgenden Abbildungen sind sogenannte „Screen Shots“, Momentaufnahmen des Bildschirms, die als Bilddatei gespeichert und dann in einem Graphikprogramm weiterverarbeitet wurden. In der Druckvorlage (A4) sind diese Abbildungen in der Regel bereits auf 75% ihrer ursprünglichen Größe reduziert, einige wenige Ausschnitte auch auf 50%.



Das Haus-Symbol, das sich auf allen Karten aller Stapel befindet (oder wenigstens befinden sollte), hat die Funktion, den Benutzer sofort zu dieser Karte, dem Navigationszentrum sozusagen, zurückzubringen, nützlich auch für den Fall, daß man glaubt, sich ‚verfranst‘ zu haben. Zurück zum Ausgangspunkt geht es also immer, ein einziger Klick reicht dazu.<sup>9</sup>

Die Karten-Metapher wird in den einzelnen Stapeln, die als Anregung und Ausgangspunkt für eigenes Gestalten mitgeliefert werden, optisch getreu umgesetzt, wie einige Beispiele zeigen mögen (vgl. **Abb. 3**). Die in das Programm integrierte Graphikumgebung erlaubt es, den Bildschirm in beliebiger Weise zu gestalten, als Karteikarte, als Notizbuch, als Zettel an der Pinwand etc. Dieses Nachempfinden realer Objekte ist durchaus gewollt und hat Methode: der Computer soll sich der dem Menschen vertrauten Welt anpassen, nicht umgekehrt. Erreicht wird auf diese Weise ein intuitiver und sicherer Umgang mit dem elektronischen Objekt.



<sup>9</sup> Die optische Entsprechung des Symbols (Haus = Home) und des Namens (Home Card) ergibt sich so nur im Englischen. In der deutschen Fassung des Programmes wurden anfänglich oft konsequent beide Elemente übertragen („Startkarte“ und ein Symbol für Orientierung), mittlerweile wird auch in Stapeln deutscher Provenienz oft das Haus-Symbol verwendet, und das aus gutem Grund: in Deutschland erstellte Stapel sind so auch in anderen Ländern unmittelbar verständlich.

Abb. 3: Karten-Beispiele

Das, was man als Benutzer auf dem Bildschirm als ein Objekt vor sich sieht, setzt sich für das Programm aus einer Reihe von Objekten in verschiedenen Schichten zusammen, die übereinandergelegt sind wie Folien (vgl. **Abb. 4**), ein Beispiel aus dem 'Help Stack', mit dem sich HyperCard selbst erklärt.

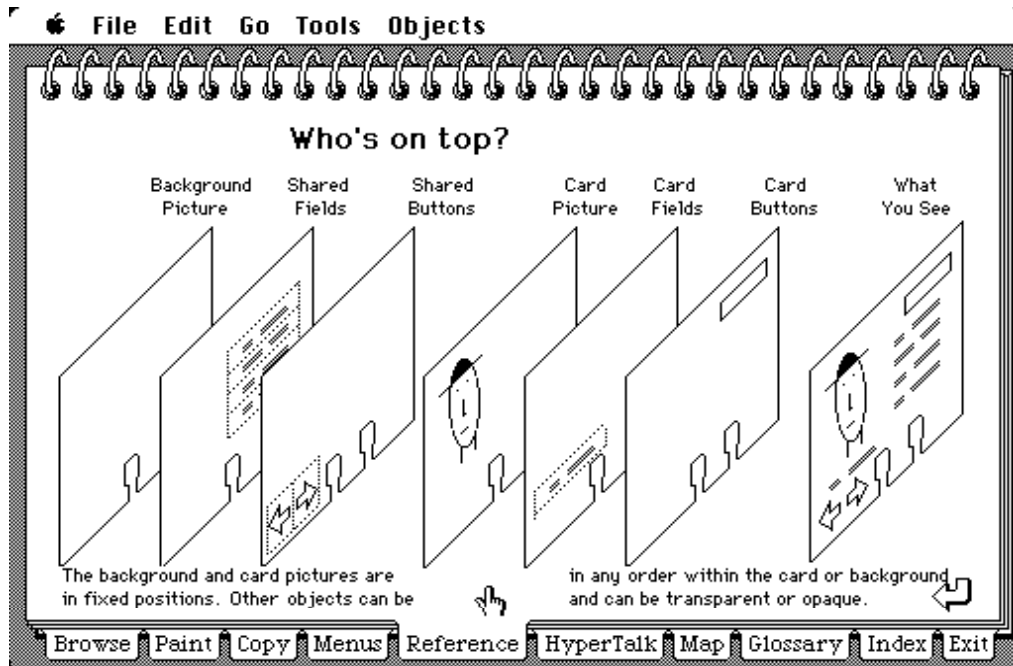


Abb. 4: Objektebenen in HyperCard

Objekte sind für HyperCard (von links nach rechts) die Karte und der Kartenstapel mit seinem Hintergrund sowie Textfelder, in denen Informationen eingetragen werden, und Knöpfe, auf die man klickt, um eine Aktion zu starten, Optionen festzulegen oder dergleichen mehr. Diese Objekte sind hierarchisch angeordnet.

So, wie eine echte Karteikarte eine bestimmte Form und ein bestimmtes Aussehen hat, wird mit dem Graphikteil des Programms der sogenannte "Hintergrund" der Karteikarten auf den Bildschirm gezeichnet (bzw. eine vorhandene Vorlage – Ideen gibt es genug – den eigenen Bedürfnissen angepaßt). Und so, wie eine Karteikarte meist vorgedruckte Linien enthält, auf die Informationen eingetragen werden sollen, können auf diesem Hintergrund die Textfelder angelegt werden, die allen Karten gemeinsam sein sollen. Wollte man z.B. den Buchbestand einer Institutsbibliothek inventarisieren, würde man hier z.B. ein Feld für den Autor, eines für den Vornamen, eines für den Titel, die Reihe, den Erscheinungsort, das Erscheinungsjahr, die Signatur usw. definieren. Auf den einzelnen Karten (eine pro Titel) würden dann später die konkreten Daten eingetragen.

Auch die Knöpfe, beispielsweise zum Durchblättern des Stapels, zur Rückkehr zur Startkarte, zur Anforderung von Hilfe oder weiterer Information etc., die stets vorhanden

sein sollen, plaziert man auf dem Hintergrund.<sup>10</sup> Die vollständigen „Menüs“ des Programmes, d.h. die Befehle, die man dem Programm geben, bzw. die Werkzeuge, die man zur graphischen Gestaltung benutzen kann, zeigt **Abb. 5**.

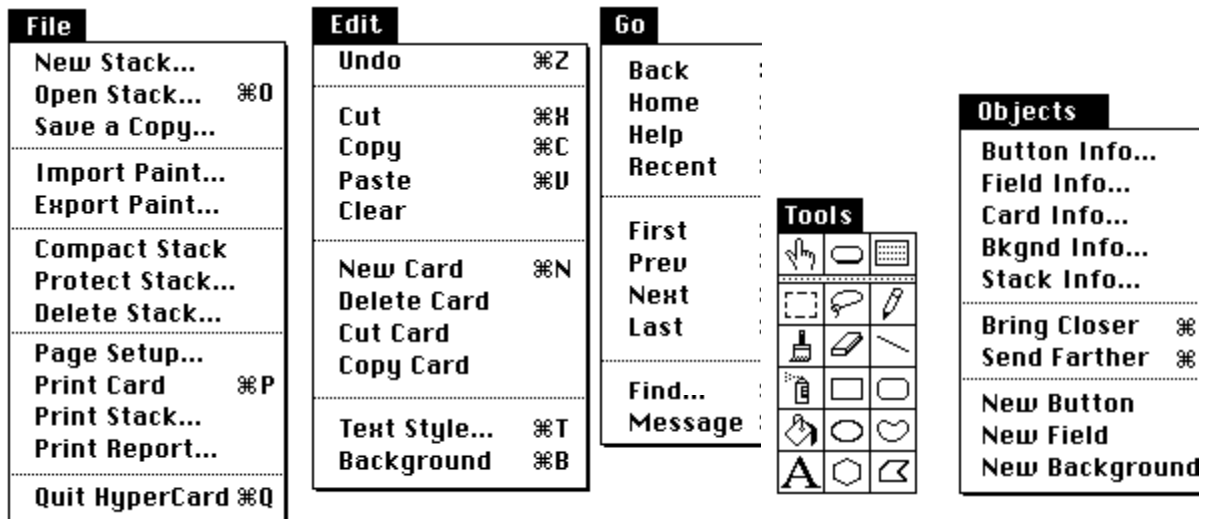


Abb. 5: Die HyperCard-Menüs

Nachdem die gemeinsamen Eigenschaften des Stapels festgelegt sind, legt man einzelne Karten an, beispielsweise für jedes neue Buch, das man eintragen will, eine neue. Auch diese einzelnen Karten eines Stapels können zu einem beliebigen Zeitpunkt noch zusätzlich individuell verändert werden, ohne daß dadurch der gemeinsame Hintergrund beeinflusst wird. Schließlich kann auch jede einzelne Karte noch graphisch verändert werden (z.B. um Teile des Hintergrundes zeitweilig auszublenden), es können spezielle Textfelder (z.B. für Kommentare) und Knöpfe (z.B. für Querverweise) angelegt werden, die dann folglich nur auf dieser Karte auftauchen.

Alle diese Schichten zusammen ergeben das Bild, das sich dem Benutzer darbietet, dem dieser Aufbau im übrigen ziemlich gleichgültig sein kann. Wichtig ist er vor allem für den Autor, also denjenigen, der einen Stapel plant und anlegt.

HyperCard ist also – auch – eine graphisch orientierte Datenbank, und schnelle Suchroutinen erlauben es, selbst größte Datenbestände in Sekundenschnelle nach einem eingegebenen Stichwort zu durchforsten. Dazu dient der Menübefehl “Find”, nach dessen Betätigung eine sogenannte „Dialogbox“ erscheint, in die man den Suchbegriff einträgt.

Ein ganz wichtiger Menübefehl, “Go Recent”, erleichtert dem Benutzer, dem beim Verfolgen von ihm interessierenden Verknüpfungen gar nicht bewußt werden muß, in welchen Kartenstapeln er gewesen oder zur Zeit gerade ist, oder wie er zu einem bestimmten Punkt zurückkommt, die Orientierung. Nach der Auswahl dieses Befehls erscheint nämlich eine Übersicht über die 42 zuletzt betrachteten Karten auf dem Bildschirm, die alle in

<sup>10</sup> Wie das konkret gemacht wird, soll an dieser Stelle nicht weiter erläutert werden; es handelt sich im Grunde einfach um die Auswahl eines entsprechenden Befehls aus dem Menü des Programmes.

entsprechender Verkleinerung gezeigt werden (vgl. **Abb. 6**). Ein Klick auf eine beliebige Karte bringt den Benutzer sofort zu dieser hin. Andere Befehle gehen zur nächsten bzw. vorigen Karte oder zur ersten bzw. letzten Karte des gegenwärtig benutzten Stapels: mit diesen Befehlen ist auch ein sequentielles Durchblättern eines Kataloges in herkömmlicher Art und Weise möglich. Ein letzter Navigationsbefehl schließlich springt zur zuletzt betrachteten Karte zurück, die sich ja ebenfalls in einem beliebigen (anderen) Stapel befinden kann und sonst womöglich gar nicht wieder aufzufinden wäre.

Die dritte Komponente von HyperCard ist, wie schon angedeutet, eine eigene Programmiersprache, HyperTalk. Mit HyperTalk wird nicht nur zum erstenmal sogenanntes „objektorientiertes Programmieren für jedermann“ ermöglicht, auch das Programmieren des Macintosh überhaupt, bisher eher nur eine Sache für Spezialisten und mit viel Einarbeitungszeit und Arbeitsaufwand verbunden, wurde damit auch den „Laien“ (und damit der Mehrheit der Benutzer) zugänglich gemacht. Die Objekte, die HyperCard kennt, sind auch die Objekte von HyperTalk, also der Kartenstapel, der Hintergrund einer Gruppe von Karten, die einzelne Karte, die Felder, die sich auf der Karte oder im Stapel generell befinden und schließlich noch Knöpfe (auf einer individuellen Karte oder im Stapel). Die Möglichkeiten von HyperTalk sind damit allerdings noch lange nicht erschöpft.

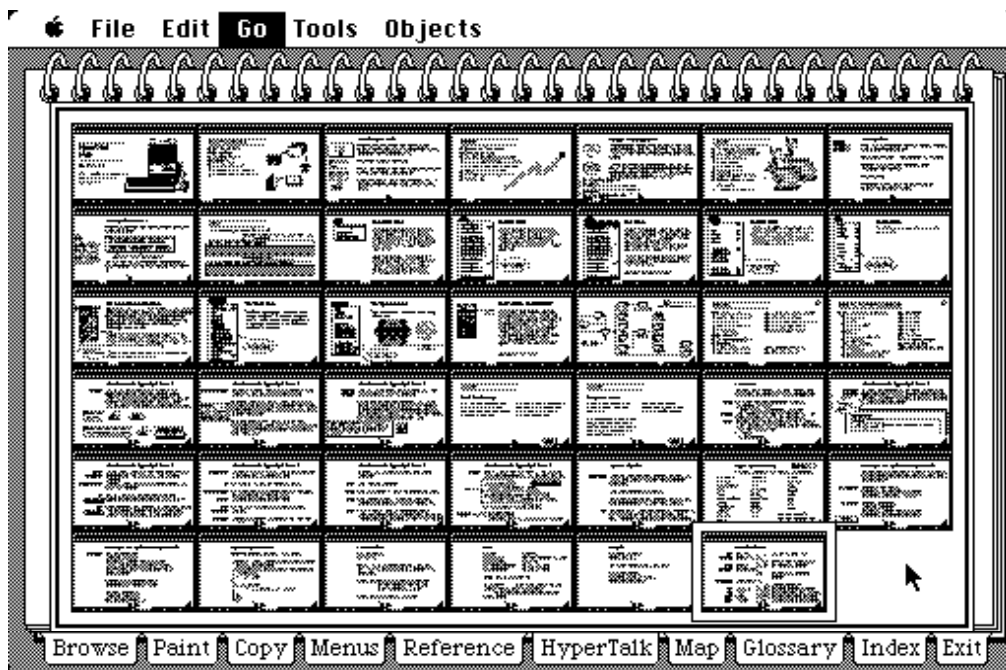


Abb. 6: "Go Recent"

Kennzeichen des mit HyperCard verbundenen objektorientierten Programmierens ist nun, daß man nicht, wie es traditionell der Fall ist, ein großes Konzept für ein Programm entwirft, also beispielsweise ein Flußdiagramm konstruiert, das alle vorkommenden Fälle in der gewünschten Weise behandelt, sondern daß kurze Programmstücke, "Scripts" genannt, an die eben genannten Objekte gekoppelt werden. Programmiert wird also nicht ein Großentwurf, sondern einzelne Reaktionen, die die Objekte je nach Benutzeraktion auslösen sollen. Ein HyperTalk-Programm setzt sich also aus vielen kleinen Bausteinen

zusammen, die gut überschaubar sind und für sich getestet werden können, kann aber selbstverständlich auf diese Weise zu einem komplexen Geflecht von Instruktionen werden, das einem Programm in einer der klassischen höheren Programmiersprachen in nichts nachsteht.<sup>11</sup>

Auf dem Bildschirm sind beispielsweise ein Links- und ein Rechtspfeil zu sehen. Bewegt man den Mauszeiger auf diese Pfeile und klickt, so bewirkt das Script, das mit diesen beiden Objekten verknüpft ist, ein Umblättern zur vorherigen bzw. zur nächsten Karte. In HyperCard existiert also gar kein zusammenhängender Programmtext mehr, sondern nur viele kleine Anweisungen, die ganz spezifische Aufgaben erledigen. Eine weitere Neuerung betrifft die Programmiersprache selbst; Anweisungen, die in dieser Sprache geschrieben werden, sehen fast aus wie English, mit englischen Ausdrücken, englischer Syntax usw. Beispielsweise ist die Anweisung "go to next card" der HyperCard-Befehl, der ein solches Weiterblättern zur nächsten Karte im Stapel bewirkt. Im dritten Teil dieses Beitrages will ich an einem Beispiel ausführlicher zeigen, wie einige ganz einfache HyperCard-Techniken genutzt werden können, um recht wirkungsvolle Effekte zu erzielen. Eine bedeutende Neuerung, auf die hier allerdings nicht weiter eingegangen werden kann, ist die Tatsache, daß diese Programmiersprache in einer ganzen Reihe von quasi natürlichsprachlichen Oberflächen existiert (Deutsch, Englisch, Spanisch, Französisch...). Jeder Benutzer einer außeramerikanischen Version programmiert in seiner Muttersprache, HyperCard selbst übersetzt den Programmtext in seine eigene Sprache (Englisch) und auch von dieser Sprache in eine andere Oberfläche. Ein deutscher Autor, der mit HyperCard in Deutsch programmiert, kann dieses Programm einem französischen Kollegen zuschicken, der auf seinem Gerät den Programmtext in Französisch angezeigt bekäme – funktionierende maschinelle Übersetzung in einem engen Spezialgebiet.

Auch wenn sich HyperCard also bemüht, „Jedermann“ das Programmieren (des Macintosh) so leicht wie möglich zu machen, muß nicht jeder, der HyperCard benutzen will, auch Programmierer werden. Studenten etwa, die Unterrichtssoftware benutzen wollen, müssen nur ganz wenige Handgriffe beherrschen, um mit dem Material sinnvoll umgehen zu können. Im Gegenteil: oft wird es sogar unerwünscht sein, daß ein Nutzer Daten verändern kann. Beispielsweise sollte ja ein Bibliothekskatalog nicht von denjenigen, die in ihm blättern, nach Belieben modifiziert werden können. In HyperCard ist auch daran gedacht worden: die Zugriffsberechtigung auf Kartenstapel kann vom Autor festgesetzt wer-

---

<sup>11</sup> Ein weiterer Unterschied besteht in folgendem: beim traditionellen Programmieren wird das Programm selber als die agierende Größe gesehen: das Programm gibt dem Benutzer beispielsweise verschiedene Optionen vor und fordert ihn auf, eine Auswahl zu treffen. Bei der Implementation einer objektorientierten Interaktion mit dem Benutzer besteht jedes Programm zunächst einmal nur aus einer „Warteschleife“. Das Programm fragt pausenlos ab, ob der Benutzer etwas tut, und wenn ja, was er getan hat, z.B. ob er eine Taste gedrückt oder mit der Maus geklickt hat, und im letzten Fall auch, wo er das getan hat, also bei welchen Bildschirmkoordinaten. Für jeden dieser Fälle reagiert das Programm mit den vorgesehenen Aktionen, weshalb man auch sagt, hier seien die Programme "event driven" („ereignisorientiert“) oder auch "user driven", weil der Benutzer jederzeit die volle Kontrolle über die Aktionen des Computers hat. Dieses Modell wird sich wegen der vielen Vorzüge, die es zu bieten vermag, mittelfristig auf allen Personal Computern durchsetzen.

den, beispielsweise auf bloßes Durchblättern, auf Löschen und Ergänzen von Einträgen etc.

Die Einsatzmöglichkeiten von HyperCard gehen aber selbstverständlich über das Erstellen von Unterrichtssoftware hinaus. Aufgrund seiner spielerischen, optisch ansprechenden Benutzeroberfläche gilt HyperCard auch als ideales Programm zur Navigation durch die Datenmengen, die die jetzt aufkommenden CD-ROM-Disks speichern können, etwa ganze Enzyklopädien, Wörterbücher und dergleichen mehr. Es ist klar, daß man Inhaltsverzeichnisse zu solchen Datenmassen dem Benutzer nicht in traditioneller Form zumuten kann, also z.B. als alphabetisch sortierte Liste aller Einträge, die man durchlaufen lassen müßte, um etwas bestimmtes nachzuschauen. Eine optische Präsentation und Gliederung von Sachgruppen führt hier offensichtlich viel schneller zum Ziel.

Auch vor HyperCard waren schon kommerzielle HyperText-Produkte erhältlich, die aber eben nur die dynamische Verknüpfung kannten (so z.B. ein Programm namens GUIDE);<sup>12</sup> erst die Kombination mit der Graphik-Komponente und der objektorientierten Programmiersprache bewirkte aber die unglaubliche Popularität der Hypertext-Implementierung von HyperCard (nicht zu vergessen die Tatsache, daß es praktisch kostenlos erhältlich ist).

Inzwischen werden einige große und ernsthafte Projekte mit HyperCard betrieben; beispielsweise gibt es 1645 Bilder der Kanadischen National Gallery of Art auf einer Videodisk-Platte, und ein HyperCard-Stapel erlaubt den Zugang zu diesem Bildarchiv, so daß Kunststudenten den Bestand beispielsweise nach Malern, Perioden oder dergleichen durchsuchen können, um die gewünschten Bilder sofort auf dem Bildschirm eines an den Computer angeschlossenen und von ihm gesteuerten Videogerätes erscheinen zu lassen. Der HyperCard-Stapel liefert zu jedem Bild die notwendigen Begleitinformationen (vgl. D'Ignazio 1988).

Ein anderes Prestige-Projekt, aus den USA, trägt den schönen Namen "The Electric Cadaver" und dient Anatomie-Studenten am Stanford Medical Center als Lehrmittel. Es erläutert Struktur und Funktion sämtlicher Teile des menschlichen Körpers; Bilder werden dabei synchron in digitalisierter Form auf dem Computerbildschirm gezeigt und in analoger Form auf einem angeschlossenen Videodisk-Gerät. Zu den über 1600 Bildern gehören Röntgenaufnahmen ebenso wie Darstellungen des Knochengerüsts, jeweils mit erläuternden Informationen. Die Zeitersparnis beim Zusammenstellen von Anschauungs-, Unterrichts- und Lernmaterial wird – nach ersten Versuchen – als ungeheuer betrachtet.<sup>13</sup>

Aber auch einheimische Projekte gibt es. An der Universität München beispielsweise wird juristisches Grundpensum in Form von HyperCard-Stapeln dargeboten (Kommentare zu Gesetzen, Fallbeispiele und dergleichen können damit auf Knopfdruck eingeblendet werden), an der Universität Hamburg werden gefährdete Vogelarten mit HyperCard katalogisiert: zu dem Bild jeden Tieres können dabei leicht seine charakteristischen Laute mit-

---

<sup>12</sup> Zu einer kurzen Besprechung vgl. etwa MICHEL (1987).

<sup>13</sup> Vgl. auch BYTE (1988/8, 14-15), MacUser 1989/3, 100-104.

abgespeichert werden, so daß sie auf Knopfdruck zur Verfügung stehen und die Bild- und Textinformation ergänzen.

### 3. Der HyperCard-Stack “Russian2”

Unter den Stacks, die bald nach der Einführung von HyperCard erschienen sind, ist auch eine Serie, die dem Russischen gewidmet ist. Autor der Stacks mit dem Namen “Learn to Read Russian” ist Martin P. Rice von der Brown-University, der auch schon früher Software für Russisch-Studenten erstellt hat, so z.B. das Programm “Verbal Aspect”, mit dem man – in Form von Einsetzübungen – den korrekten Aspektgebrauch üben kann. Während dieses Programm noch in der traditionellen Weise erstellt wurde, d.h. der Programmtext nach den Regeln einer der klassischen Programmiersprachen, in diesem Falle Pascal, geschrieben und dann von dem sogenannten Compiler in ein lauffähiges Programm übersetzt wurde, nutzen die neuen Stacks die Möglichkeiten von HyperCard und HyperTalk. Es ist im übrigen sicher kein Zufall, daß dieses Projekt an der Brown University realisiert wird (s.o.).

Das Übungsmaterial besteht bis jetzt aus sieben Teilen, den zwei einführenden Stacks “Learn to Read Russian” und “Read More Russian”, in denen in erster Linie die Aussprache einzelner Laute sowie ganzer Wörter geübt wird, sowie fünf aufeinander aufbauenden Lektionen. Diese Stacks sind, wie der Autor mitteilt, Grundlage eines geplanten “stack-book for the study of Russian”.

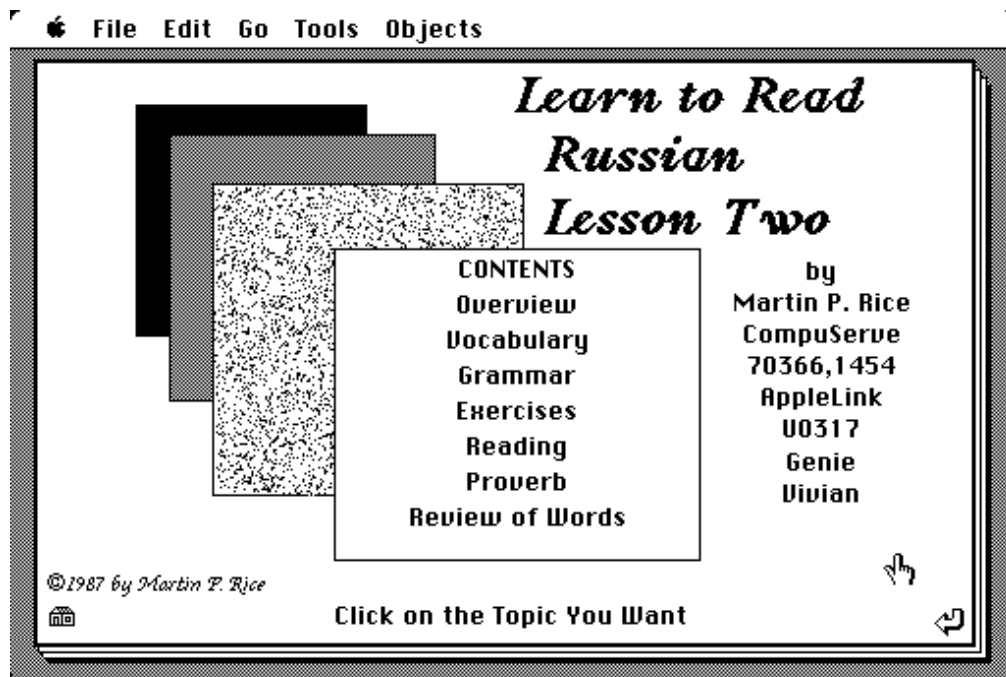


Abb. 7: Eröffnungskarte

Bei diesen Stacks handelt es sich um sogenannte “Public Domain Software”, d.h. sie können auf nichtkommerziellem Wege beliebig weitergegeben werden, ohne damit die Urheberrechte des Autors zu verletzen. Kommerzielle Software ist inzwischen ebenfalls auf dem Markt.<sup>14</sup>

Ich will im folgenden versuchen, Struktur und Arbeitsweise der zweiten Lektion, stellvertretend für die übrigen, zu verdeutlichen. Diese zweite Lektion bietet eine Einführung in die Kasus des Russischen, speziell Nominativ und Akkusativ, und in die Konjugation der Verben, garniert mit einigen Zutaten, wie Lese- und Übersetzungsübungen, Vokabeltraining und anderes mehr.

Der Stack besteht insgesamt aus 31 Karten – eine Tatsache, die dem Benutzer an keiner Stelle bewußt zu werden braucht. Wenn man die Lektion startet, gelangt man zur ersten Karte, der Schaltzentrale für die ganze Lektion. Von hier aus springt man zu den verschiedenen Übungen, hierher kehrt man jeweils stets zurück. **Abb. 7** zeigt diese Eröffnungskarte mit dem Titel des Stacks und dem Hinweis auf den Autor.<sup>15</sup> Aus dem Inhaltsverzeichnis wählt man einen der Übungsteile aus, indem man den Zeiger auf das betreffende Wort bringt und die Maustaste drückt; dies bewirkt jeweils einen Sprung zur ersten Karte des betreffenden Übungsteils, von denen manche nur aus einer einzigen Karte bestehen, andere aber auch aus einer ganzen Reihe von Karten.

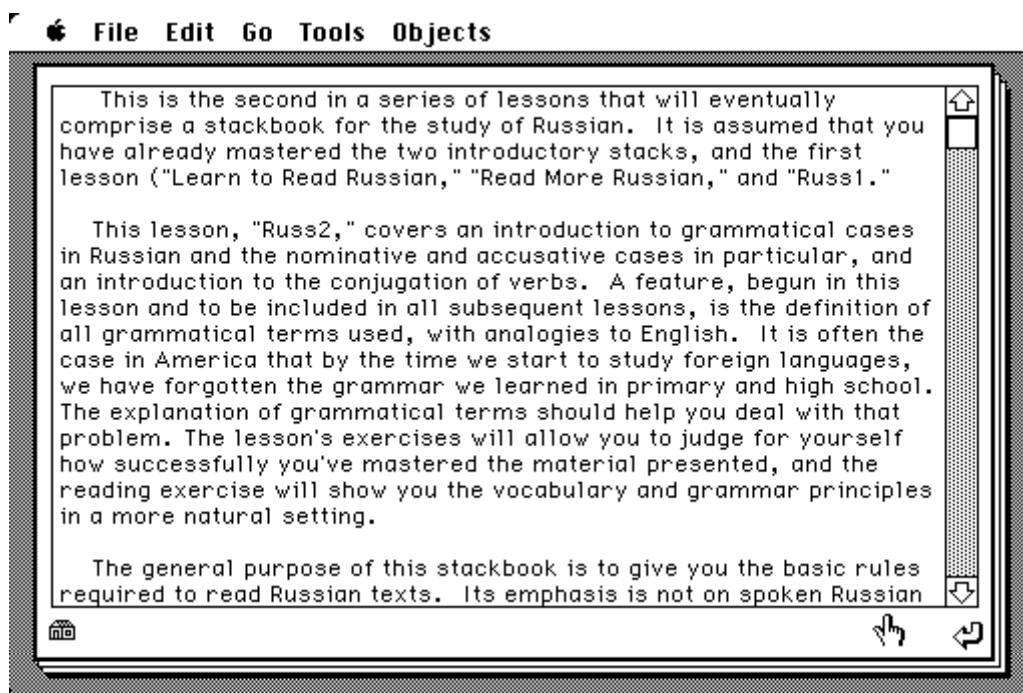


Abb. 8: Karte “Overview”, Fenster mit Rollbalken

14 Von der Firma “HyperGlott Software” in Knoxville/Tennessee sind folgende Stacks erhältlich: ‘Introduction to Russian’, ‘Russian Noun Tutor’, ‘Russian Verb Tutor’, ‘Word Torture Vocabulary Drill in Russian’, jeweils ca. \$20-30. Sie haben dem Autor noch nicht vorgelegen, so daß eine Beurteilung ihrer Qualität an dieser Stelle nicht erfolgen kann.

15 Abb. 18 zeigt die Struktur der Stapels in symbolisierter Form; die Übersicht ist im Original nicht Bestandteil des Stapels, sondern wurde von uns hinzugefügt.



Wenn man z.B. auf das Wort “Overview” klickt, dann springt man zu einer Karte, die generelle Informationen über diese Russisch-Stacks enthält, das Lernziel, das mit den Stacks verfolgt wird (Lesefähigkeit, nicht Sprechfähigkeit) und einige weitere Hinweise, vgl. **Abb. 8**.

Der Text der allgemeinen Informationen ist länger, als – auch bei kleiner Schrift – auf einer Karte unterzubringen wäre, weshalb er in einem sogenannten „Fenster“ auf der Karte erscheint. Bei Betätigung des „Rollbalkens“ (dies ist die Leiste am rechten Rand) kann man den restlichen Text ebenfalls lesen. Ein Klick auf das “Return”-Symbol in der rechten unteren Ecke bringt den Benutzer zur Eröffnungskarte von **Abb. 7** zurück. Zwischen beiden Karten besteht also eine einfache Verknüpfung, die ein Hin- und Herspringen zwischen ihnen erlaubt. Die Befehle, die dies ermöglichen, sind äußerst einfach; beim Klicken auf den Knopf “Return” beispielsweise wird folgende Befehlssequenz abgearbeitet (“Contents” ist der Name der ersten Karte):

```
on mouseUp
  go to card "Contents"
end mouseUp
```

Eine derart simple Verknüpfung zwischen zwei Karten kann in HyperCard menügesteuert hergestellt werden, die Software schreibt dann das Programmstück, das hier “Script” heißt, selbst. “on mouseup” ist eine der in HyperTalk eingebauten Anweisungen und heißt: „wenn die Maustaste gedrückt und wieder losgelassen wurde“; dies ist das Ereignis, auf das der Rest des Scripts reagiert. Die Syntax der eigentlichen Anweisung (Zeile 2) ist dabei – für eine Programmiersprache – nicht sehr restriktiv: man könnte genauso gut “go to first card”, “go to card number one” oder dergl. schreiben, aber die gewählte Form hat den Vorzug, unabhängig von der tatsächlichen Abfolge der Karten im Stapel und auch selbsterklärend zu sein.

Von einer ähnlich einfachen Struktur sind noch zwei weitere Übungsteile des Stapels, nämlich “Reading” und “Proverb”. Beide bieten aber eine zusätzliche Technik und demonstrieren weitere Möglichkeiten. Wenn man auf “Proverb” klickt, springt man zur gleichnamigen Karte, die in **Abb. 9** wiedergegeben ist. Ein Klick auf das invers dargestellte Sprichwort bringt selbiges zu Gehör. Unter dem russischen Text liegt nämlich unsichtbar eines der in HyperCard möglichen Objekte, ein „Knopf“. Das Script dieses Buttons bewirkt das Abspielen der Aufnahme. Denn darum handelt es sich hier: eine ganz normale Tonbandaufnahme wurde in digitalisierter Form im vorliegenden Stapel unter dem Namen “Povtorenie” gespeichert, und diese Aufnahme wird in diesem Augenblick nun „abgespielt“; es handelt sich also nicht um eine Sprachsynthese, sondern um eine Digital-Analog-Wandlung.<sup>16</sup>

---

<sup>16</sup> Eine echte Sprachsynthese als Erweiterung des Betriebssystems gibt es auch, ist aber vorläufig nur für das Englische implementiert und bewußt einfach gehalten. Die Digitalisierung von Tonbandaufnahmen oder auch direkter Aufnahmen über ein Mikrofon ist mit dem „MacRecorder“ möglich, einem zigarettenschachtelgroßen, recht preisgünstigen Kästchen, das als Bindeglied zwischen Tonband oder Mikrofon und

Gleichzeitig passiert auf dieser Karte noch etwas: es taucht vor dem Hintergrund der Karte ein neues Feld auf, das die Übersetzung des Sprichwortes enthält. Wie z.B. eine Folie über eine andere gelegt wird, so erscheint hier diese Information vor dem augenblicklichen Hintergrund; nur der tatsächlich unter dem Feld liegende Teil der Karte wird dabei verdeckt, der Rest bleibt sichtbar, vgl. **Abb. 9**. Das Sichtbarwerdenlassen der Übersetzung ist mit dem Anhören des Sprichwortes gekoppelt. Das Script dieses Buttons lautet vollständig folgendermaßen und ist wohl selbsterklärend:

```
on mouseUp
  show card field "Translation"
  play "Povtorenie"
end mouseUp
```

Der Autor hat also – speziell auf dieser Karte – ein Feld mit dem Namen “Translation” eingerichtet, das auf diese Weise gezeigt wird. Voraussetzung dafür, daß dies funktioniert, ist natürlich, daß jedesmal beim Sprung zu dieser Karte das Übersetzungsfeld zunächst unsichtbar ist, was mit dem folgenden Karten-Script, das natürlich bei der vorausgehenden Benutzung aktiv wird, geschieht:

```
on closeCard
  hide card field "Translation"
end closeCard
```

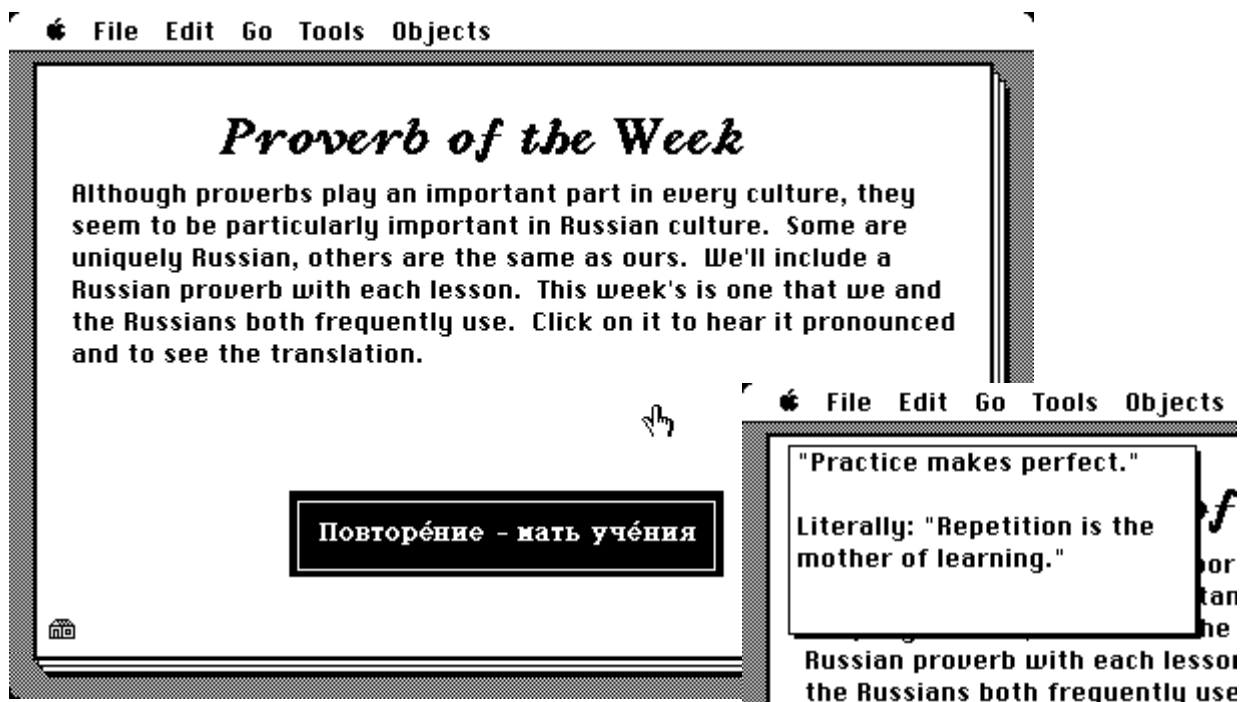


Abb. 9: Proverb-Karte (mit Overlay)

Computer fungiert. Mit der mitgelieferten Software erfolgt die Digitalisierung und Speicherung der Aufnahme im Computer. Die so gespeicherten Sound-Dateien stehen dann wiederum in HyperCard (oder anderen Programmen) zur Verfügung. Die Software dieses „MacRecorder“ ist durchaus auch für einfache phonetische Analysen geeignet, denn sie erlaubt es, die Aufnahme mit großer Präzision zu segmentieren, einzelne Schnipsel abzuspielen, sie zu manipulieren etc.

Alternativ kann man das Übersetzungsfeld sofort wieder verschwinden lassen, wozu man einmal hineinklickt. Auch für diese Aktion reicht ein simples Script, das mit dem Übersetzungsfeld verknüpft ist (man kann also nicht irgendwo auf der Karte klicken):

```
on mouseUp
  hide card field "Translation"
end mouseUp
```

Soviel zu dieser Karte.

Der zweite Übungsteil, der ebenfalls aus einer Karte besteht, ist “Reading”. Auch hier finden wir wieder Overlays, die beim Betätigen des Info-Knopfes bzw. beim Anklicken eines Satzes sichtbar werden, vgl. **Abb. 10**. Ansonsten werden nur jetzt schon bekannte Techniken verwendet.

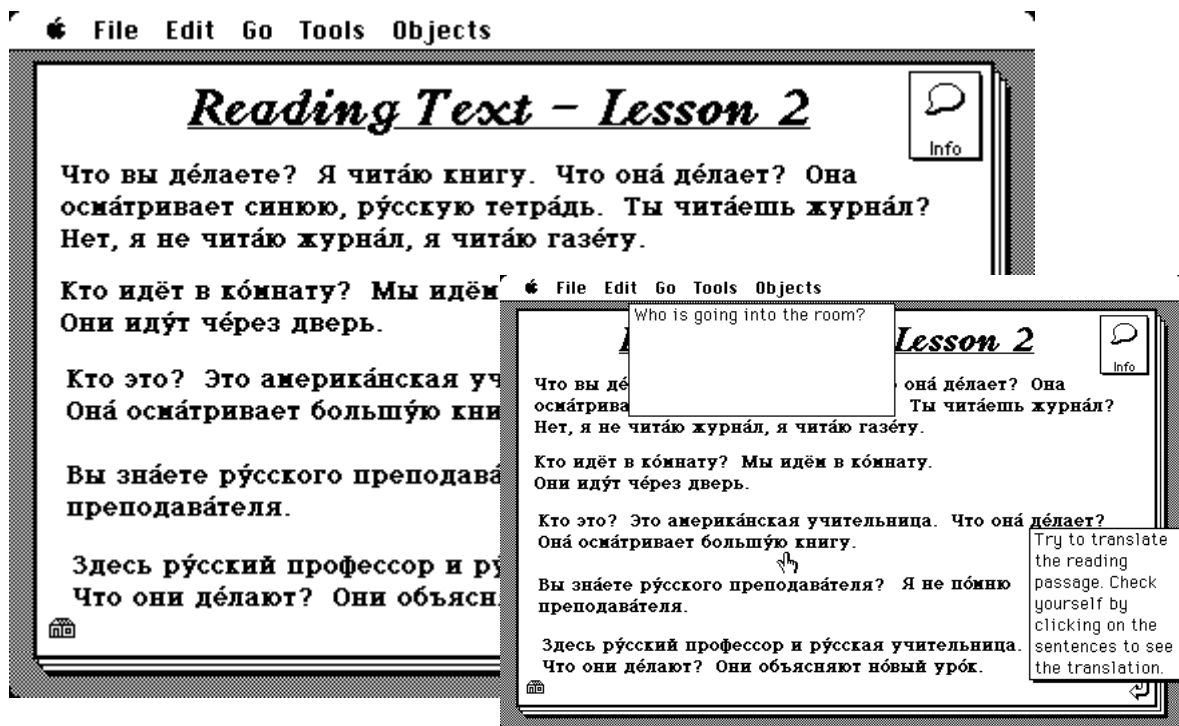


Abb. 10: “Reading” mit Overlay (rechts)

Ein weiterer Übungsteil, “Vocabulary”, ist im Prinzip ganz ähnlich strukturiert. Ein Klick auf das Wort “Vocabulary” bringt einen zur Vokabelkarte, die in **Abb. 11** wiedergegeben ist, gleich mit dem Overlay, das erscheint, wenn man auf “Info” klickt. Unten rechts das Feld, das die englische Übersetzung des gerade ausgewählten Wortes zeigt. Wie bei dem oben erläuterten “Proverb” ist auch hier jedes Wort mit einem – unsichtbaren – Knopf unterlegt, mit dem Übersetzung und Aussprachedemonstration gekoppelt sind. Das Script des ersten Wortes lautet beispielsweise:

```
on mouseUp
  play "Amerikanskij"
  put "American" into card field "Answer"
  show card field "Answer"
end mouseUp
```

Was der Knopf “Hear them all” bewirkt, erklärt sich wohl von selbst. Zu einigen Vokabeln gibt es Anmerkungen, die auf einer eigenen Karte versammelt sind; genausogut hätte man jede Fußnote in einem dafür vorgesehenen Feld vor dem Hintergrund dieser Karte zeigen können.

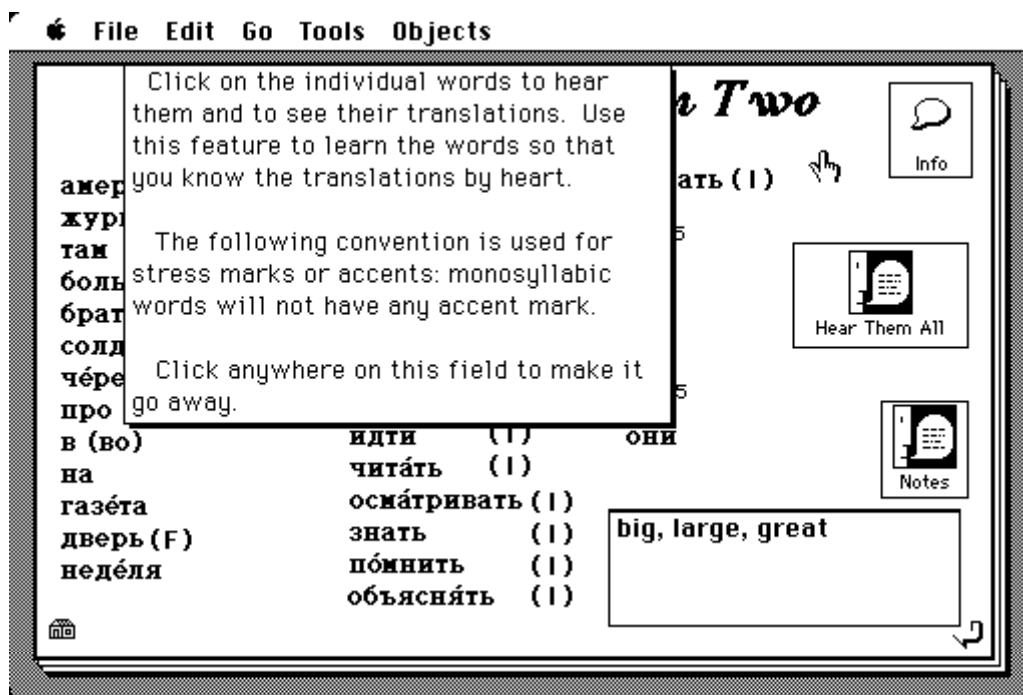


Abb. 11: Vokabelkarte

Etwas anders ist der “Review of Words” benannte Teil strukturiert. Auch er besteht aus zwei Karten, die in diesem Falle aber hierarchisch sozusagen auf gleicher Stufe stehen und nur sequentiell abgearbeitet werden sollen. **Abb. 12** zeigt die erste der beiden Karten; ein Klick auf die Windrose oben rechts bringt einen zur zweiten Karte und von dort wieder zurück zur ersten..

Eine weitere Technik lässt sich mithilfe des Übungsteils “Exercises” demonstrieren, nämlich der Dialog mit dem Benutzer. Es erscheint zunächst die Karte der **Abb. 13**, deren Inhalt man nur zur Kenntnis nehmen soll, um dann zur eigentlichen Übung weiterzugehen. Im ersten Teil besteht die Aufgabe darin, morphologisch-syntaktisch zusammenpassende Wortformen als solche zu erkennen.

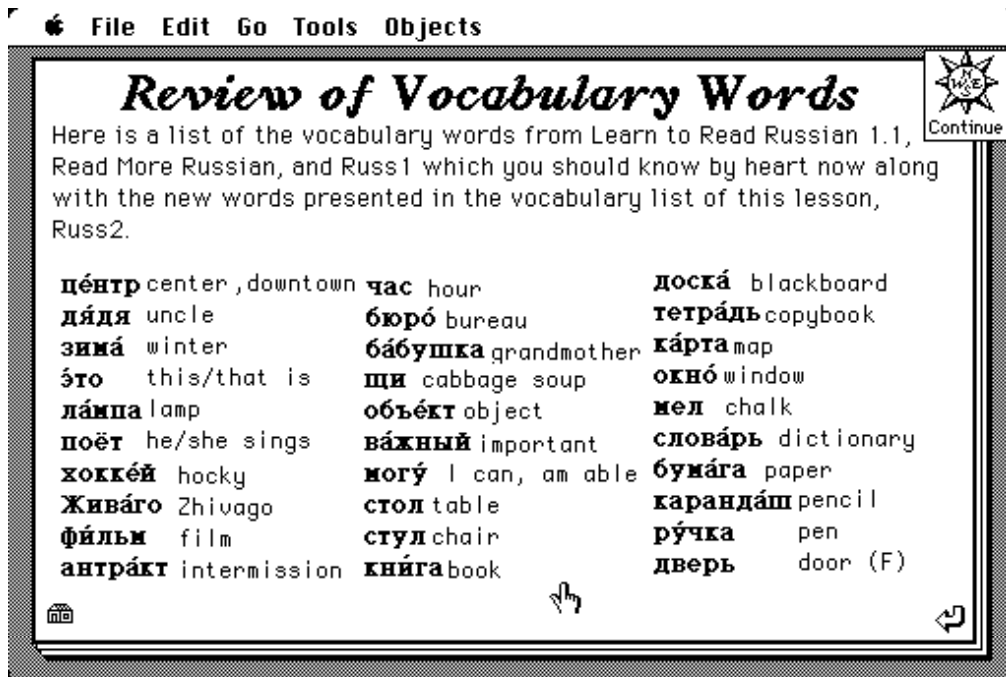
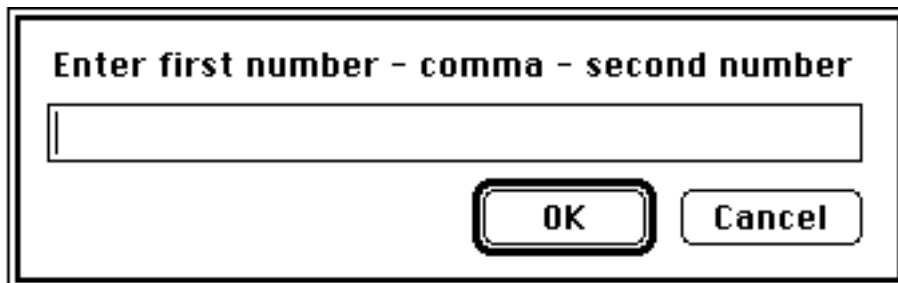


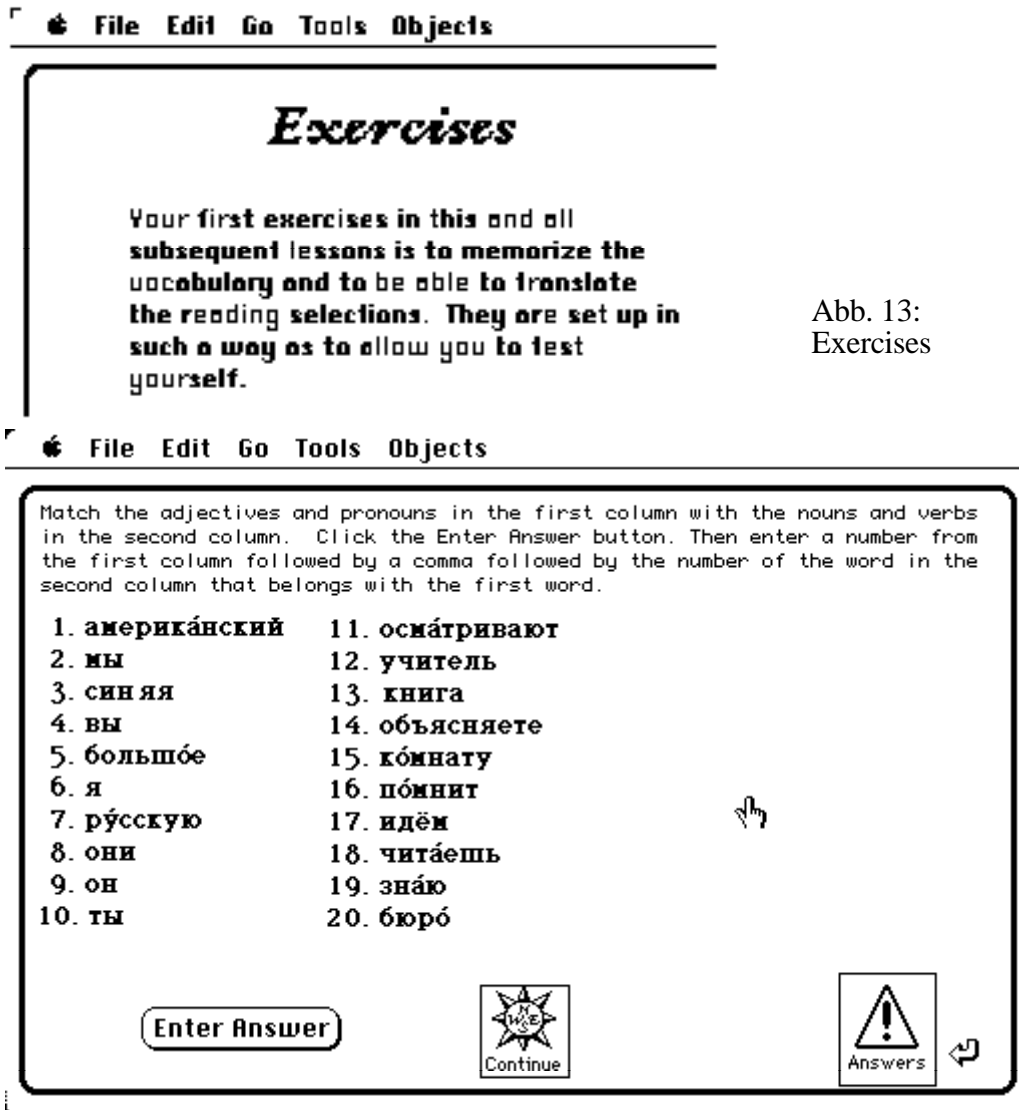
Abb. 12: "Review of Words"

Die Bereitschaft zur Eingabe einer Antwort zeigt man dem Programm an, indem man auf den "Enter Answer"-Knopf drückt. Daraufhin erscheint – wieder vor dem Hintergrund der Karte, den ich hier allerdings ausgeblendet habe –, die folgende Dialogbox:



Mit einem Klick auf "O.K." bestätigt man die Vollständigkeit der Eingabe, mit einem Klick auf "Cancel" bricht man ab. In beiden Fällen verschwindet die Dialogbox wieder, und sofern man auf O.K. gedrückt hat, erscheint die Beurteilung der Eingabe, die man durch Klicken auf den Knopf zu quittieren hat:





Auch der Aufwand, solche interaktiven Übungen zu konstruieren, ist sehr gering. Nur die eigentliche Anweisung muß expliziert werden, das „Drumherum“, wie die eingerahmte Dialogbox mit den beiden Knöpfen, wird von HyperCard automatisch erzeugt. Das Script des Enterknopfes lautet daher eigentlich nur:

```
on mouseUp
  ask "Enter first number - comma - second number"
end mouseUp
```

Die Eingabe des Benutzers wird automatisch in einen sogenannten "Container", d.h. eine Variable, eingelesen, die einfach "it" heißt, so daß die Abfragen, ob die Eingabe richtig oder falsch war, folgende allgemeine Form haben ('<>' steht für 'not equal', 'RW' für 'Right/Wrong'):

```
if first item in it is [value] and second item in it <> [value]
then put "F" into RW
```

Die Variable RW enthält die Beurteilung der Antwort; sie wird zu Beginn einfach auf "T[true]" gesetzt, und alle falschen Antworten schreiben stattdessen ein "F[false]" hinein. Diese Abfrage muß natürlich für jede Antwort einmal gesondert formuliert werden; am Ende kann dann aber die generelle Anweisung kommen:

```
if RW "F" then answer "That is wrong!"
                else answer "That is right!"
end if
```

Einige weitere Abfragen dienen noch dazu, Eingaben, die ganz außerhalb des möglichen Bereiches liegen, also Tippfehler oder Mißverständnisse, von den Antworten zu trennen, die grammatisch falsch sind. Die Programmierarbeit desjenigen, der solche Arten von Übungen erstellen will, ist also wirklich minimal, und die Anweisungen, die zu schreiben sind, lesen sich (fast) wie normales Englisch.

Auf der zweiten Karte dieses Teils geht es um das Einüben von Flexionsformen; das Muster der Interaktion mit dem Benutzer ist aber das gleiche (vgl. **Abb. 14**). Hier, wie auch auf der ersten Karte, kann man sich die korrekten Antworten durch Anklicken des "Answer"-Knopfes zeigen lassen.

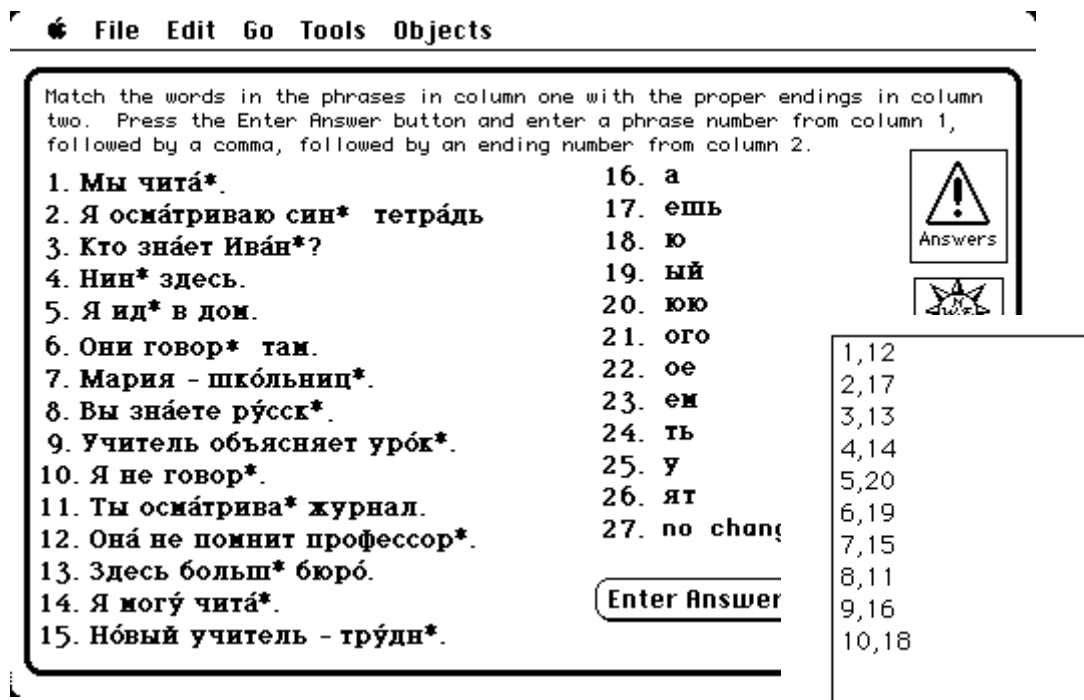


Abb. 14: Übung zur Morphologie

Der komplexeste Übungsteil beschäftigt sich mit der Grammatik des Russischen. Zur Auswahl stehen Themen aus dem Nominal- wie Verbalbereich, vgl. **Abb. 15**. Daran, daß hier im Inhaltsverzeichnis so viel Platz gelassen ist, erkennt man, daß die gleiche Karte offenbar in anderen Lektionen – mit nur geringen Modifikationen – ebenfalls zur Auswahl unter den jeweiligen Themen gedacht ist.

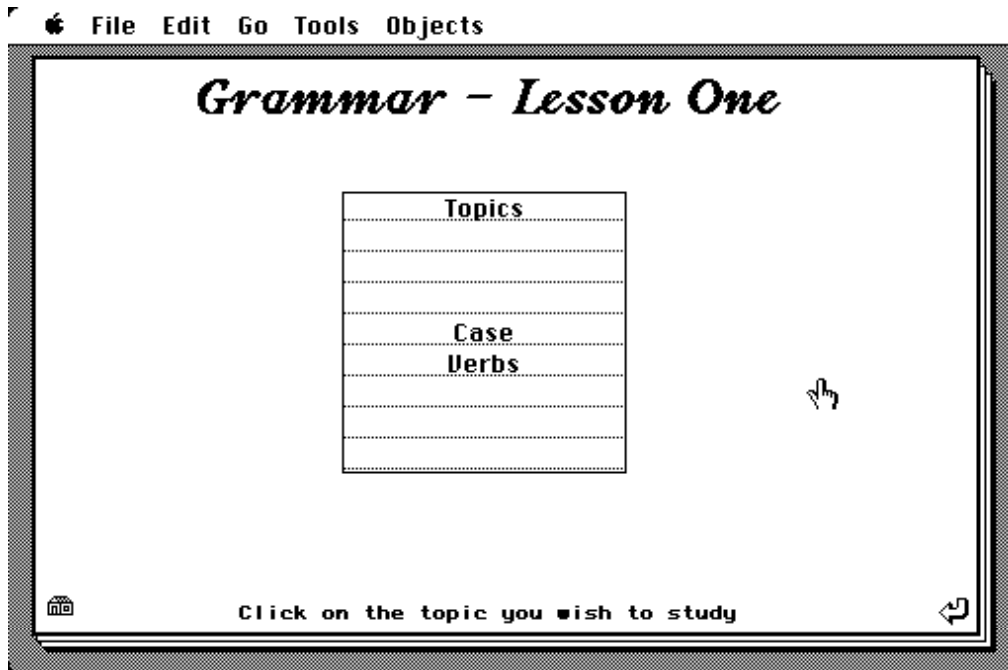


Abb. 15: Inhaltsverzeichnis des Grammatik-Teils

Die Techniken, die in diesem Übungsteil verwendet werden, sind nunmehr schon bekannt. Die Auswahl führt zunächst zu einer der beiden nachstehend abgebildeten Karten (Abb. 16a-b):

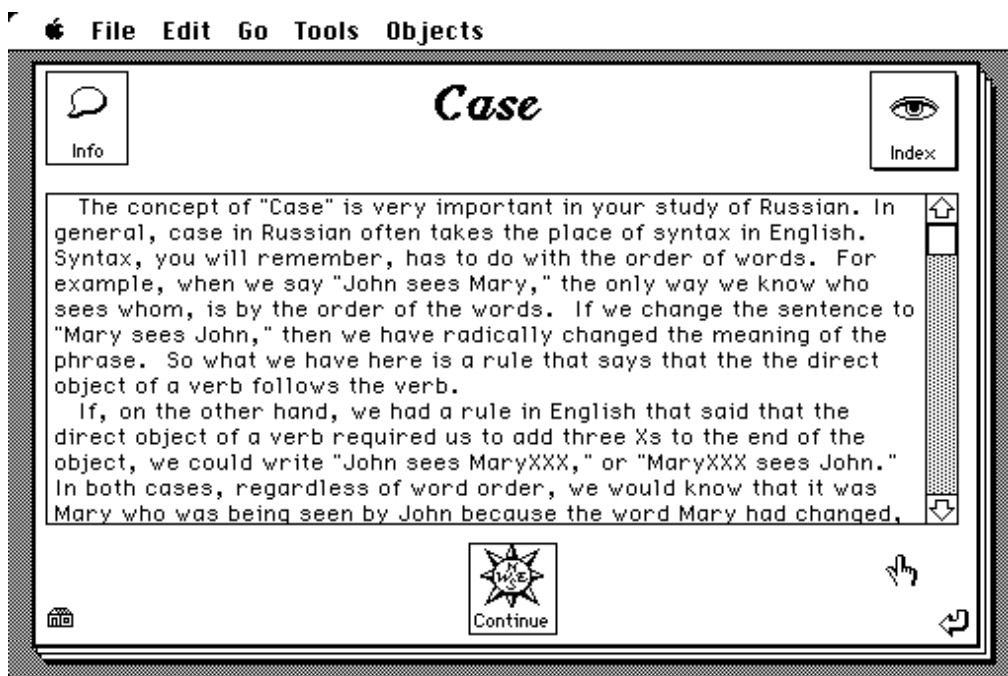


Abb 16a: Einleitung des Grammatik-Teils



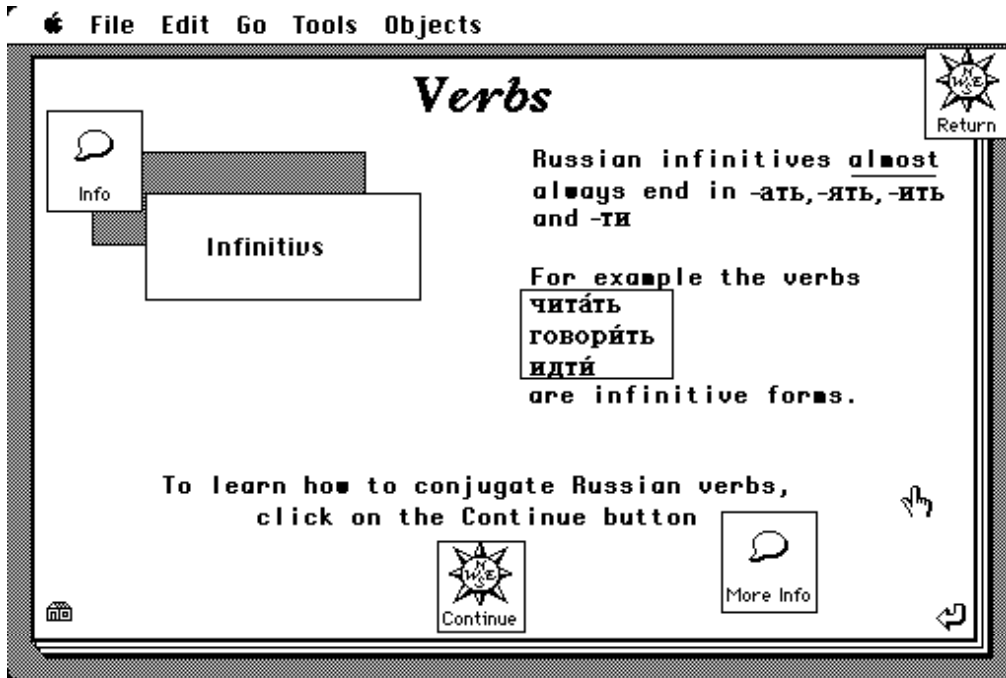


Abb. 16b: Einleitung des Grammatik-Teils

Der Teil zu den Nomina ist ziemlich komplex, und ich will mich deshalb darauf beschränken, hier noch abschließend den Verb-Teil kurz vorzustellen. Nachdem man die Informationen der ersten Karte zur Kenntnis genommen hat und sich auch die weiterführenden Erläuterungen hat zeigen lassen, geht es mit einem Klick auf "Continue" weiter zur zweiten Verbkarte, die in **Abb. 17** wiedergegeben ist. Die beiden unteren Felder zeigen dann noch jeweils die erste bzw. die zweite Konjugation.

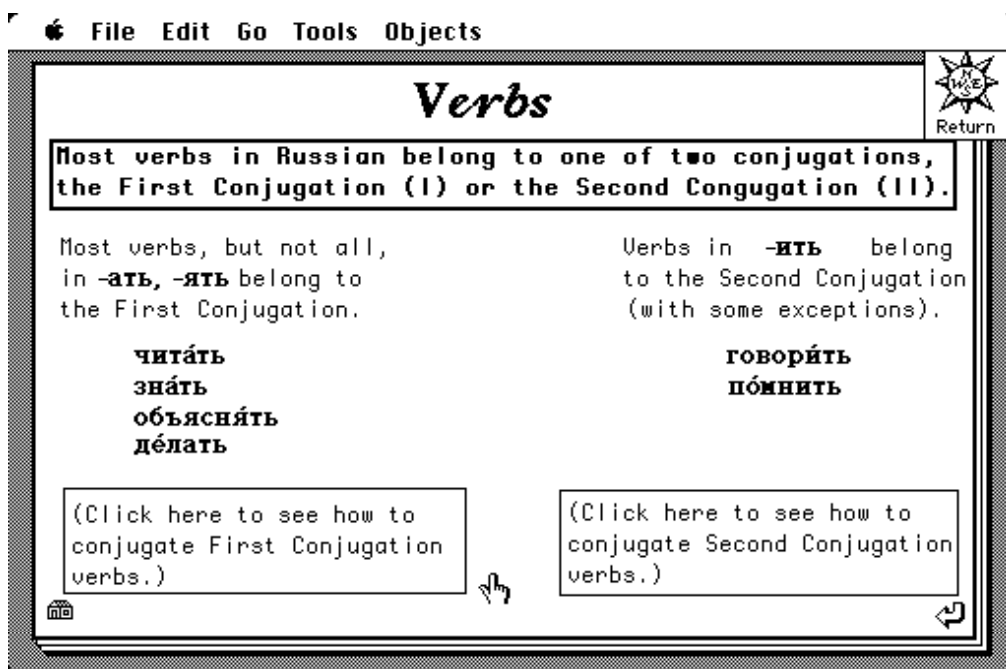


Abb. 17: Verb-Karte

Perfekt ist der Stack, den ich hier vorgestellt habe, aus Benutzersicht jedoch nicht: an der graphischen Gestaltung kann man offensichtlich das eine oder andere auszusetzen haben. So wie *Lehrbücher* eines professionellen Layouts bedürfen, profitieren auch HyperCard-Stacks von einer kompetenten graphischen Gestaltung. Vor allem fehlt dem Stack eine generelle Orientierungskarte, die dem, der glaubt, sich „verfranst“ zu haben, jederzeit zeigt, wo er sich befindet. Eine solche Orientierungskarte kann beispielsweise so aussehen, wie in der nachstehenden Abbildung (Abb. 18), der die entsprechende Karte des HyperCard-“Help“-Stacks als Vorlage dient. Das Erstellen eines solchen strukturellen Diagramms, das die Verknüpfungen der Karten untereinander im Überblick zeigt, gehört zu den Empfehlungen für gute Hypertext-Systeme. Es macht zugleich einen der häufiger geäußerten Kritikpunkte an diesen Systemen gegenstandslos.

Soviel zum Inhalt dieses Russisch-Stacks und zur Analyse der Techniken, mit denen der Stapel erstellt wurde. Ich hoffe, es ist deutlich geworden, daß sich HyperCard in der Tat hervorragend als Autorensystem zum Erstellen derartiger Übungen nutzen läßt.

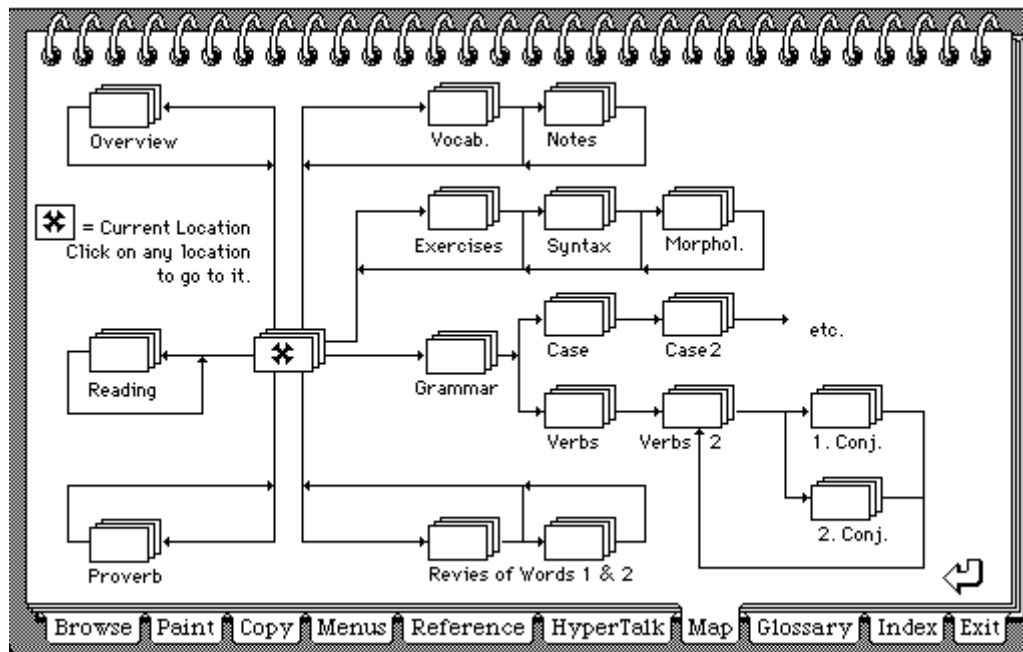


Abb. 18: Orientierungskarte

#### 4. Perspektiven, Anwendungsmöglichkeiten

Mit ein bißchen Phantasie kann sich jeder Lehrende wie Lernende sicher eine ganze Reihe von möglichen Anwendungen für HyperCard im Slawistischen Unterricht denken. Ich will nur einige nennen, die mir im Augenblick als besonders interessant erscheinen:

- Ein HyperCard-Stack zur *Phonetik* des Russischen. Hier könnte beispielsweise ein Vokaldreieck erscheinen, in dem man jeden Vokal mitsamt seinen positionsbedingten Allophenen auf Knopfdruck hören könnte, auch kontrastiv deutsch-russisch. Die Verlage-

rung der Artikulation von unbetonten Vokalen könnte in einer einfachen Animation sichtbar gemacht werden, ebenso die Akkomodation der Vokale in den vier relevanten Positionen (vor/ nach harten/weichen Konsonanten), etwa so, wie sie im Lehrbuch von OLIVERIUS (1974) dargestellt ist. Artikulationsdiagramme, wie sie bei GABKA (1975) kontrastiv zu finden sind, könnten wie Folien übereinander gelegt werden, so daß man die Veränderungen in der Artikulation einzelner Laute leicht nachvollziehen kann.

- Für einen Stack zur *Intonation* des Russischen hätte man beispielsweise das Ton-Material von BRYZGUNOVA (1969) als Ausgangspunkt zur Verfügung und könnte die zugehörigen Übungen entsprechend umsetzen.

- In der *Syntax* könnte man HyperCard benutzen, um etwa Satzkonstruktionen durchsichtig zu machen, indem ein komplexer Satz aus einer einfachen Struktur erzeugt wird, wobei der Effekt syntaktischer Regeln hier wirklich nachvollziehbar gestaltet werden könnte, wenn der Lehrende oder ein Student auf Knopfdruck zwischen der expandierten und der kondensierten Version der Darstellung eines Satzes hin- und herschalten kann. Oder man könnte auf einzelne Regeln zeigen und deren Auswirkungen auf einen Satz studieren. Hierbei wäre HyperCard nicht nur Demonstrationsmedium, sondern könnte auch heuristische Funktion übernehmen, denn bei komplexen Transformationen kann man immer noch am einfachsten optisch kontrollieren, ob das tatsächliche mit dem gedachten Ergebnis übereinstimmt.

- Im *lexikalischen Bereich* könnte eine Hypertext-Version von ZALIZNJAKs “Грамматический словарь русского языка. Словоизменение” (1977) von großem Wert sein: man müßte dann nicht mehr jedes Kürzel der Einträge im Grammatikteil nachschlagen, sondern könnte auf Knopfdruck sofort die Musterparadigmen einblenden und wieder verschwinden lassen. Auch die Regeln zur Erzeugung der einzelnen Formen, denen ja ein exakter Algorithmus zugrundeliegt, könnten sofort am gewünschten Wort demonstriert werden. Oder man stelle sich das HyperCard-Äquivalent zu DAUM/SCHENK vor: zu jedem Verb sind sofort seine Formen sichtbar, regelmäßige Bildungen könnten von unregelmäßigen auch optisch unterschieden werden.

- In der *historischen Grammatik* könnte man sich beispielsweise einen Stapel vorstellen, der mit einer einfachen Animationstechnik zeigt, wie aus dem Phonemsystem des Gemeinostslawischen schrittweise durch Spaltung und Zusammenfall der Phoneme das Phonemsystem des heutigen Russisch entstanden ist (beispielsweise als Umsetzung der Darstellung von BORETZKY 1984). Oder wie sich durch phonologische, morphologische und morphologische Veränderungen die Flexionsparadigmen von ihrer altrussischen Form zur heutigen entwickelt haben (regelgestützte Computerprogramme für diesen Zweck gibt es im übrigen bereits, vgl. BORIN 1988).

- Oder man denke an das *Altkirchenslawische*: Generationen von Studenten bestimmen an immer wieder den gleichen Texten immer wieder die gleichen Formen. In einem HyperCard-Stapel könnte jedes Wort auf Knopfdruck Informationen über seine Form oder die Grundform (Nominativ, Infinitiv) selbst enthüllen, und diese Informationen würden nur sichtbar, wenn sie wirklich gebraucht und gewünscht werden. Auch die Übersetzung ganzer Sätze könnte auf Wunsch gezeigt werden. Wer diese Informationen nicht braucht, kann den Text lesen oder wenigstens in Angriff nehmen, ohne von der Lösung sofort be-

einflußt zu werden. Eine (Selbst-)Kontrolle, wie oft welche Hilfe in Anspruch genommen wurde, ließe sich leicht realisieren.

- Auch im *landeskundlichen Unterricht* könnte HyperCard mit Erfolg eingesetzt werden, man denke nur an die Analyse von Gesten, von Werbung und dergleichen, Gebieten also, bei denen optische und sprachliche Information sehr eng zusammengehören. In HyperCard ist es problemlos möglich, wie schon deutlich geworden ist, Bild und Ton miteinander zu verknüpfen. Bildvorlagen können dabei mit einem sog. Scanner leicht von einer Vorlage abkopiert und in HyperCard verwendet werden.

Alle diese Ideen sind im Rahmen des traditionellen Programmierens nicht zu verwirklichen, da sie einfach zu viel Arbeit erfordern und es keinen „Markt“ für solche Produkte gibt. Mit HyperCard aber kann auch jemand, der sich mit dem Programmieren im traditionellen Sinne niemals beschäftigen würde, schon Hilfsmittel für den Unterricht erstellen, die bisher nicht realisierbar waren. Darin liegt die große Chance, die sich mit HyperCard eröffnet.

Natürlich sind die Einsatzmöglichkeiten, die ich hier kurz skizziert habe, für deutsche Universitäten gegenwärtig mangels entsprechender Geräteausstattung in der Regel noch Zukunftsmusik. HyperCard kann aber nicht nur für das Selbststudium sinnvoll eingesetzt werden, genauso könnte schon ein einziges Gerät mit entsprechenden HyperCard-Stapeln als Hilfsmittel im Unterricht zur schnellen Demonstration eingesetzt werden, zumal sich der Inhalt des Bildschirms mit einem entsprechenden Zusatzdisplay auch über einen normalen Overhead-Projektor einem größeren Kreis zugänglich machen läßt.

#### LITERATUR

Begemann, M.L., Conklin, J.:

1988 The Right Tool for the Job. *Byte* 10, 255–266.

Boretzky, N.:

1984 Immanente Geschichte der russischen Sprache. In: H. Jachnow (Hg.), *Handbuch des Russisten*. Wiesbaden, 404-443.

Borin, L.:

1988 A Computer Model of Sound Change: An Example from Old Church Slavonic. *Literary and Linguistic Computing* 3/2, 105-108.

Брызгунова, Е.А.:

1969 Звуки и интонация русской речи. Лингафонный курс для иностранцев. Москва.

Bush, V.:

1945 As we may think. *Atlantic Monthly* 176/1, 101-108. [Reprint in Lambert, S., Ropiequet, S. (eds.), *CD-ROM. The New Papyrus*. Redmont 1986.]

[Byte]

1988 Hypertext. What it is—and isn't. How it works. How you can use it. *Byte* 10, 234-268.

[CD-ROM Review]

1988 HyperMedia: Greater Than The Sum of Its Parts. *CD-ROM Review* Vol. 3, No. 3 (May), 18-27.

[Communications of the acm]

1988 *Communications of the acm*, Special Issue: Hypertext. Vol. 31, No. 7.

Custer, L.:

1987 Stacking the Deck. *MacUser* 12, 112-127.

Conklin, J.:

1987 Hypertext: An Introduction and Survey. *IEEE Computer* 20/9, 17-41.

van Dam, A.:

1988 Hypertext '87: Keynote Address. *Communications of the acm* 31/7, 887-895.

Daum, E., Schenk, W.:

1976 Die russischen Verben. Grundformen • Aspekte • Rektion • Betonung • Deutsche Bedeutung. 10. Auflage. München.

D'Ignazio, F.:

1988 The National Gallery of Art in Your Computer. *Compute* 4, 48.

Engelbart, D.:

1963 A conceptual framework for the augmentation of man's intellect. In: Howerton, P.W., Weeks, D.C. (eds.), *Vistas in information handling: Vol. 1. The augmentation of man's intellect by machine*. Washington, 1-29.

Fiderio, J.:

1988 A Grand Vision. *Byte* 10, 237-244.

Franklin, C.:

1988 An Annotated HyperText Bibliography. *Online. The Magazine of Online Information Systems* 12/2, 42-46.

Frisse, M.:

1988 From Text to Hypertext. *Byte* 10, 247-253.

Gabka, K. (Hg.):

1975 Die russische Sprache der Gegenwart, Bd. 1: Einführung in das Studium der russischen Sprache, Phonetik und Phonologie. Leipzig.

Goodman, D.:

1987a The Two Faces of HyperCard. *MacWorld* 10, 122-129.

1987b The Complete HyperCard Handbook. New York 1987.

Halasz, F.G.:

1988 Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the acm* 31/7, 836-855.

Крысин, Л.П.:

1974 Русский язык по данным массового обследования. Опыт социально-лингвистического изучения. Москва.

Leemon, Sh.:

1988 The Hazards of HyperCard. *Compute* 4, 49.

MacUser

1985ff. MacUser. The Macintosh Resource. Boulder/Colorado.

Michel, St.:

1987 Guide – A Hypertext Solution. *CD-Rom Review* July/August, 22-24.

Nelson, T.H.:

1974 Computer Lib. Chicago.

1981 Literary machines. Swathmore.

Oliverius, Zd.F.:

1974 Vvedenije v fonetiku ruskogo jazyka. Lekcii i upražnenija. Praha.

Raymond, D.R., Tompa, F.W.:

1988 Hypertext and the New Oxford Dictionary. *Communications of the acm* 31/7, 871-879.

Smith, K.E.:

1988 HyperText – Linking to the Future. *Online. The Magazine of Online Information Systems* 12/2, 32-40.

Thornburg, D.D.:

1988 The Power of HyperCard, Part 1–2. *Compute* No. 1, 25–27, No. 2, 54-55.

Williams, G.:

1987 HyperCard. *Byte* 12, 109-117.

[Wheels for the Mind]

1987 HyperCard: How One Good Idea Can Lead to Another. In: *Wheels for the Mind* 3/4, 1-20.

Young, J.S.:

1986 Hypermedia. *MacWorld* 3, 116-121.

Zalznjak, A.A.:

1977 Грамматический словарь русского языка. Словоизменение. Москва.