



[www.allegro-c.de/allegro](http://www.allegro-c.de/allegro)

## Wann, wenn nicht winters ...

*Winter works its way to wrap  
all the land in icy covers,  
biting noses,  
freezing roses,  
rather not a time for lovers -  
though for thoughts, a welcome gap...*

... wäre die Zeit - die Ernte unter Dach und Fach - Ergebnisse zu sichten, Konzepte zu überdenken, Pläne neu zu schmieden? Denn das nächste Frühjahr kommt bestimmt, und es soll einen nicht mit neuen Herausforderungen kalt erwischen. Werfen wir also einen Blick auf die Bilanz, prüfen wir, was mit dem Erreichten anzufangen ist, schauen wir, was zu tun sein wird im nun anbrechenden letzten Jahr mit der "19" und vorletzten des Jahrhunderts.

Der in Jahresfrist erfolgende Übergang von der 19 zur 20 ist in den letzten Monaten zum Stoff für Horrorszenarien bis hin zu Untergangsvisionen geworden. In der Tat, wenn irgendwo in einer Folge von Aktionen ein Programm nicht von 1999 auf 2000 schaltet, sondern von 99 auf 00, kann das unabsehbare Konsequenzen haben, z.B. Aufforderungen, die seit 99 Jahren überfälligen Bücher doch nun endlich zurückzubringen, mit astronomischen Verzugsgebühren... Ein scheinbar triviales Problem macht drastisch deutlich, was gern verdrängt wird: Ein winziger Schritt für den menschlichen Verstand kann sich für den Rechner als unüberbrückbare Kluft erweisen - wie ernüchternd! Natürlich *kann* er in viel größeren Zahlen "denken", was er aber nicht kann, ist *mitdenken*, d.h. von sich aus merken, wo etwas nicht stimmen kann, also Sinn von Unsinn in den ihm aufgetragenen Aktivitäten unterscheiden. Denn jede Software ist ein höchst unvollkommenes Abbild einer Realität, die in ihrer Gesamtheit für die künstliche Intelligenz noch immer viel zu komplex ist. Das gilt nicht nur für den Kalender, es gilt insbesondere für jede Art von Datenbank, also auch für OPACs und für die Suchmaschinen im Internet. Darin Systeme zum Beantworten von Fragen zu erblicken, ist eine grobe Fehleinschätzung, sie arglosen Nutzern so vorzustellen, demnach Irreführung. (Es sei denn, man erklärt immer ganz genau, was eine "Frage" und eine "Antwort" ist und was nicht! Die Definition wird jedoch vom Alltagsverständnis dieser Wörter stets weit entfernt sein.) Wir tun gut daran, dies bei aller Freude über neue Programme mit neuen Leistungen immer einmal zu bedenken. Vielleicht an langen Winterabenden.

Das **Jahr-2000-Debakel** wird *allegro*-Anwendern jedoch erspart bleiben (immer vorausgesetzt, das Betriebssystem macht mit). Zwar kann auch *allegro* nicht mitdenken, aber es hat schon immer die Jahreszahlen vierstellig behandelt. Nur das Ausleihprogramm *aLF* tat dies an einigen Stellen nicht, doch wurden diese Stellen inzwischen bereinigt; das aktualisierte Programm, auf der Basis **V15e**, wurde allen Anwendern bereitgestellt. Die besagten Unsinnsmahnungen wird es also nicht geben. Probleme, jedoch keine gravierenden, können jetzt höchstens noch im Bereich der Parametrierung auftreten. Dazu geben wir in dieser Nummer noch weitere Hinweise (s. Seite 13), die wir zur aufmerksamen Beachtung empfehlen.

Nun aber zur Bilanz der Entwicklungsarbeit des Jahres 1998:

## Von Z39.50 bis *alcarta*

"Neue Schnittstellen für *allegro*" stand als Titel über dem Projektantrag an die DFG, mit dem 1996 die Entwicklung einer Z39.50-Schnittstelle angeschoben wurde. Auf 18 Monate war diese Aufgabe zunächst geschätzt worden. Es zeigte sich dann aber bald, daß man damit nicht auskommen würde, obwohl ein sehr kompetenter Entwickler gefunden wurde. Die DFG genehmigte den Verlängerungsantrag um 12 Monate, die nunmehr abgelaufen sind. Im Oktober, auf dem Expertentreffen '98, konnten das realisierte Konzept und das Serverprogramm vorgestellt werden. Seitdem ist es im Einsatz und hat von den Testern bereits positive Noten erhalten. Die Projektaufgabe ist also erfüllt, erreicht wurde aber weit mehr als "nur" die Erstellung einer Z39.50-Schnittstelle. In dieser Nummer wollen wir Bilanz ziehen und Zusammenhänge aufzeigen.

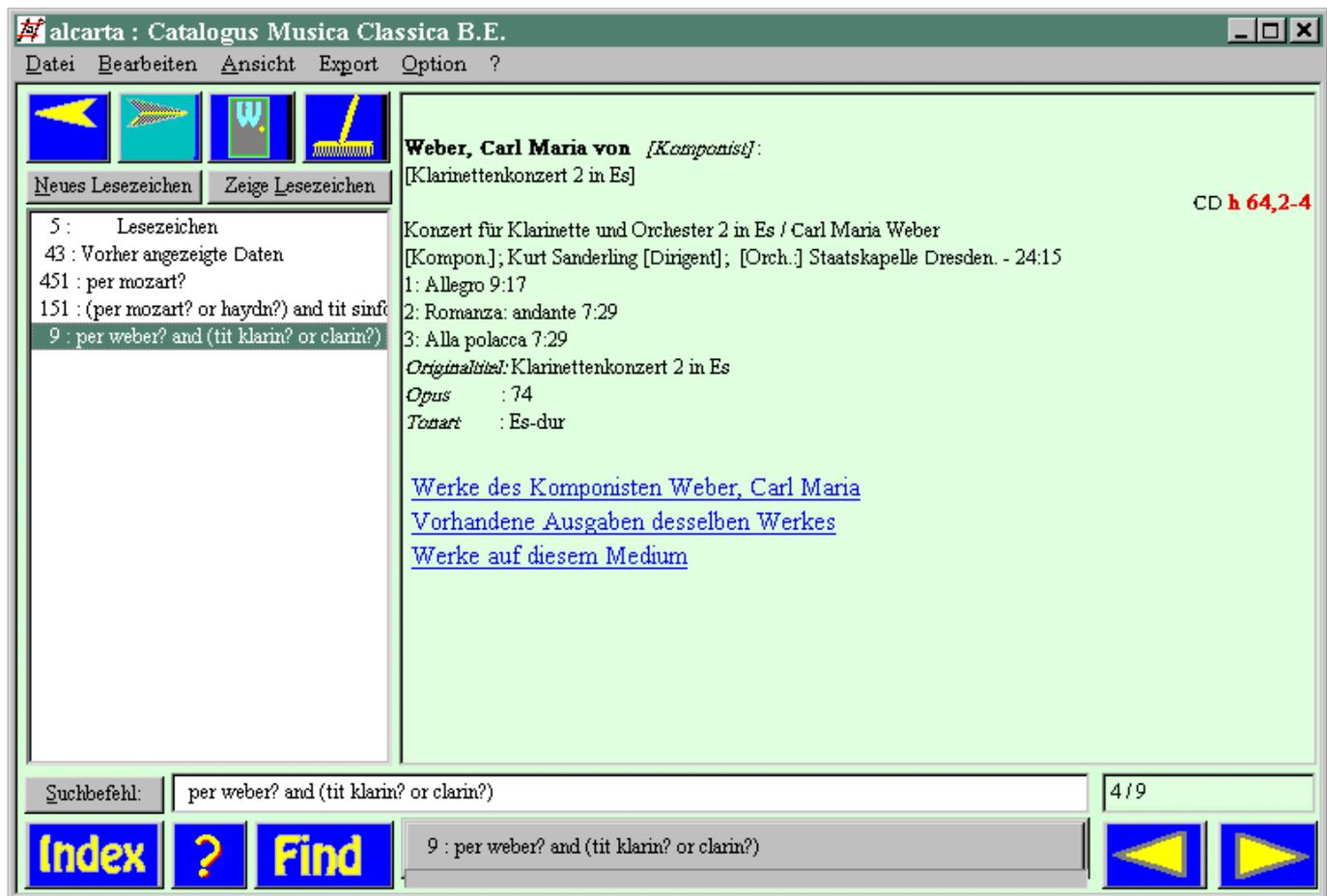
Wenn man *allegro*-Datenbanken mit Z39.50 nach außen öffnen will, braucht man nicht eines, sondern zwei Serverprogramme, die sich die Arbeit teilen. Das erste und zuerst entwickelte ist der *avanti*-Server (*news* 43), der die lesenden und schreibenden Zugriffe durchführt. Dieser Server beruht auf der Klassenbibliothek in C++ (*news* 39). Das zweite ist der Z39.50-Kommunikationsserver und beruht auf der DBV-OSI-Programmbibliothek. Wie diese beiden Server zusammenwirken, wird ab Seite 8 dargestellt. Wichtig ist: der *avanti*-Server kann auch zugleich ganz andere Zugriffsverfahren unterstützen, die mit Z39.50 nichts zu tun haben. Zahlenmäßig überwiegen dabei die WWW-Anwendungen, die in der Regel auf einer Kombination von HTML-Seiten und Perl-Skripten beruhen. Über 40 Web-Kataloge arbeiten bereits mit dieser Methodik (*news* 44; eine Liste findet man unter [www.allegro-c.de/allegro/ac-dbs.htm](http://www.allegro-c.de/allegro/ac-dbs.htm)). Geeignet ist sie auch für den Einsatz im Intranet und mit Programmiersystemen wie Delphi. Beispiele wurden auf dem Expertentreffen gezeigt.

Die Serverfunktion (auch "Target" genannt) ist aber nur die eine Seite der Medaille. Das Z39.50-Konzept hat auf der Client-Seite (sog. "Origin"-Funktion) zum Ziel, dem Endbenutzer das Suchen in fremden Datenbanken zu gestatten, ohne daß er die gewohnte Umgebung verlassen muß. Ein Suchbefehl, der in der lokalen Datenbank zu nichts führt, soll weitergereicht werden können an andere Systeme, wobei man, so das Konzept, von deren andersartigen Befehls- und Datenstrukturen nichts wissen muß, nur Z39 müssen sie "können". Die besondere Komplikation für *allegro* lag darin, daß das "alte" OPAC-Programm gar nicht befehlsorientiert arbeitet. Die Zugriffe laufen ja immer über Register, in denen man blättert, und logische

Kombinationen werden auf vergleichsweise unorthodoxe Weise durchgeführt. Nun ist das Unorthodoxe bisweilen keineswegs das Schlechtere, aber fast alle wollen eben doch das Konventionelle (ohne zu hinterfragen, warum es zu einer Konvention wurde und ob diese die bestmögliche Lösung ist oder es jemals wirklich war). *Fazit*: es mußte ein anderes OPAC-Programm her, eins mit Suchbefehlen und Boolescher Logik, wie man es z.B. von Systemen wie Pica kennt. Die Vorarbeiten dazu gehen ebenfalls auf die Entwicklung der Klassenbibliothek zurück: als erstes entstand 1995 das Versuchsprogramm VP (*news* 40), mit dem man Befehle geben konnte wie z.B. `find per schiller? and tit räuber`. Nun aber kam hinzu, daß die "alten" *allegro*-Programme, die unter DOS laufen, durch Windows-Programme abgelöst werden müssen. Also war ohnehin ein neuer Oberflächenansatz zu machen, da konnte gleich der Einbau eines Find-Befehls mit Boolescher Logik vorgesehen werden, zusätzlich zum Zugriff über die Register, der unbedingt erhalten bleiben sollte. Diese neue Oberflächenentwicklung war dann zu kombinieren mit einem Z39.50-Zugriff nach außen. Gegen Ende der Projektzeit, und ganz am Ende der Erntezeit '98, kam auch diese neue Entwicklungslinie zu einem ersten Resultat: das Programm *alcarta* wurde im Dezember 1998 vorgestellt. Es bildet den anderen Schwerpunkt in dieser Ausgabe. Für die Mehrheit der Anwender ist dieses wohl die am dringendsten erwartete Neuerung: ein OPAC-Programm für Windows. *alcarta* ist aber mehr: man sollte es als "Datenbank-Browser" bezeichnen. Warum, und was das heißt, steht im nachfolgenden Bericht.

## *alcarta* : Datenbanken à la carte OPAC und Datenbank-Browser für Windows

Statt langer Worte zuerst eine Abbildung. So erscheint das Hauptfenster des neuen Programms:



Man sieht hier, wie gerade eine *bolero*-Datenbank benutzt wird. Mit dem Suchbefehl (siehe dazu S.14)

`per weber? and (tit klarin? or clarin?)`

wurde eine Ergebnismenge gebildet. Diese besteht aus 9 Einträgen, der vierte davon wurde ausgewählt und ist zu sehen. Das Erscheinungsbild des Programms und seine Funktionsweise orientiert sich an Netscape, dem bekannten Internet-Browser, und das mit gutem Grund: Unter WWW-Nutzern ist kaum ein anderes Programm so bekannt, dadurch findet man sich mit diesem neuen OPAC schnell zurecht. Zugegriffen wird hier nicht auf Internet-Adressen (oder HTML-Dokumente), sondern auf Datensätze einer *allegro*-Datenbank, aber ansonsten gibt es viele Ähnlichkeiten. Die wichtigsten sind:

- Die blau unterstrichenen Elemente können angeklickt werden, um andere Daten hervorzuholen (z.B., siehe oben, [Werke des Komponisten ...](#)), aber auch, um externe Programme zu starten oder Hilfetexte zu öffnen.
- Die Buttons links oben mit den Pfeilen blättern rückwärts und vorwärts in den vorher angezeigten Datensätzen.

- Man kann einzelne Datensätze mit einem Lesezeichen versehen und die so markierten Sätze jederzeit als Ergebnismenge anzeigen lassen.

### Wie bekommt man das Programm?

*alcarta* ist, wie APAC (und wie der Acrobat-Reader, s.u.), kostenlos und kann auch mit eigenen Datenbanken zusammen vertrieben werden, um diese benutzungsfertig z.B. auch auf CD anbieten zu können. Alle Abonnenten erhalten es neben *a99* etwa im April '99 auf der nächsten CD-ROM, es ist aber schon jetzt Internet zugänglich. Man steuere folgende Seite an:

[www.allegro-c.de/allegro/alcarta](http://www.allegro-c.de/allegro/alcarta)

dann erfährt man alles Weitere. Auch mehrere Beispieldatenbanken liegen dort bereit. *a99* liegt nicht dort, sondern im Bereich AC15 auf dem FTP-Server. Dieser Bereich ist nur registrierten Abonnenten mit Passwort zugänglich.

### Datenbank-Browser?

Web-Nutzer kennen das "Acrobat"-Konzept der Firma Adobe: Texte im PDF-Format kann jeder mit hohem Komfort lesen und ausdrucken (im Gegensatz etwa zu WinWord-Dokumenten), weil ein Leseprogramm, der Acrobat-Reader, kostenlos erhältlich ist. Nur der *Produzent* des Textes braucht ein lizenzpflichtiges Programm, mit dem er PDF-Dateien erstellen kann.

Etwas Ähnliches hatte die UB Braunschweig im bibliothekarischen Datenbankwesen schon vor längerer Zeit realisiert: das OPAC-Programm APAC für DOS wurde zur Freeware erklärt. Dadurch konnte jeder *allegro*-Anwender seine Datenbank beliebig verbreiten, ohne daß jeder Endnutzer sich eine Lizenz aus Braunschweig hätte besorgen müssen. Nur der Datenbank-*produzent* braucht die (nicht besonders teure) Lizenz für das Kernsystem. Objekte wie der "*berliner allegro-Catalog*", der "*Kirchliche Verbundkatalog*", die "*Theologische Literaturdokumentation*" oder auch kleinere wie die "*Lippische Bibliographie*" etc. konnten deshalb zu erschwinglichem Preis auf CD-ROM realisiert werden. Die UB Braunschweig hat gleichfalls ihren OPAC auf CD herausgebracht und kann die Scheibe für DM 5,- kostendeckend abgeben. Kleinere Objekte wie die Formate-Datenbank oder die UNICODE-Zeichensatz-Datenbank, wurden zum Herunterladen bereitgestellt.

Wie gesagt, das Programm *alcarta* hat wegen seiner äußerlichen und funktionalen Ähnlichkeit mit dem bekannten Web-Browser den Vorteil, schnell erlernbar zu sein. Es ermöglicht eine Datenübernahme in Windows-Standardsoftware auf hohem Niveau (incl. Sonderzeichen, Farben, Schriftattributen). *alcarta* ist also ein Datenbank-Browser, der sich am bewährten "Look and Feel" der Web-Browser orientiert - soweit dies sinnvoll ist. Es ist jedoch eigenständig, nicht als HTML/CGI-Lösung realisiert, sondern entscheidend effizienter, d.h. spürbar schneller, und auch leichter zu konfigurieren.

Die für die Gestaltung von Anwendungen entscheidende Parametrierung ist vollständig dokumentiert. Für die Datenanzeige und Hilfstexte wird nicht HTML benutzt, sondern RTF. Das hat große Vorteile, die wir hier nicht im Einzelnen darstellen. Wer auf HTML schwört, kann einen Katalog mit der Web-Methodik und *avanti*-Server auch im Intranet und auf Einzelplatzrechnern realisieren! Jedoch langsamer und umständlicher und nicht mit allen Funktionen, die *alcarta* hat. Geht es aber darum, eine "lebende", ständig aktuelle Datenbank unter's Volk zu bringen, dann ist *alcarta* nur für den lokalen Gebrauch richtig, dann braucht man für den online-Zugriff über's Netz die WWW-Methodik mit *avanti*.

(Beispiele: [www.allegro-c.de/allegro/ac-dbs.htm](http://www.allegro-c.de/allegro/ac-dbs.htm) )

Ein Datenbankproduzent kann praktisch jede beliebige, selbstgestaltete *allegro*-Datenbank, so wie sie ist, verfügbar machen. Notwendig ist dafür die Erstellung geeigneter Parameter für die Anzeige der Datensätze sowie evtl. bestimmter Hilfsseiten. Dabei helfen die dokumentierten Beispiele, die wir zum Herunterladen per FTP anbieten, und eine Checkliste der notwendigen Dateien (siehe S. 6/7).

Weil *allegro* kompakt speichert, können relativ große Datenbanken zum Herunterladen im Netz bereitgestellt werden. Das gesamte Paket, das eine Datenbank ausmacht, muß dann nur auf ein Unterverzeichnis entpackt werden. Mitzuliefern ist eine INI-Datei, die die notwendigen Einstellungen für *alcarta* enthält, und schon ist die Datenbank benutzbar. Bei sehr großen Datenbanken (etwa > 200.000 Sätze) bleibt die Verbreitung auf CD-ROM eine weitere Möglichkeit: *alcarta* kann auch direkt auf eine CD zugreifen. Dann ist der lokale Speicherbedarf Null. Dieselbe Datenbank ist, wie gesagt, auch immer noch mit dem DOS-Programm lesbar. (*alcarta* selbst braucht weniger als 1 MB Plattenspeicher).

*Nochmals*: das neue Konzept ergänzt, aber ersetzt nicht die schon verbreitete Methodik, eine Datenbank online im WWW anzubieten, was mit dem *avanti*-Server und neuerdings zusätzlich mit dem Z39.50-Server geschehen kann. Nicht jeder Datenbank-Produzent hat Zugang zu einem Web-Server, und nicht jede Datenbank braucht unbedingt den Web-Zugriff, außerdem ist eine lokale Nutzung erheblich schneller und komfortabler als eine Nutzung über WWW. In Zukunft gibt es also drei Methoden zur Auswahl, eine *allegro*-Datenbank zu publizieren.

Wer Win'95 oder NT noch nicht hat, kann immer noch auf dieselben Datenbanken auch mit dem "alten" Programm APAC zugreifen, das bis zur Stunde in zahlreichen Bibliotheken die OPAC-Funktion ausfüllt, in Novell- oder NT-Netzen.

*Beispieldatenbanken*: An der genannten Adresse liegen ZIP-Pakete mit Beispieldatenbanken zu den Formaten *bolero*, HANS, USMARC, NRW-MAB, außerdem die Formate-Datenbank, eine SWD-Teilmenge, Regensburger Systematik und andere.

Für die Entwicklung steht als nächste Aufgabe an, einige Elemente aus *alcarta* auch in *a99* einzubauen, vor allem das RTF-Anzeigefeld. Beide Programme sollen ferner die Fähigkeit erhalten, auf mehr als eine Datenbank zuzugreifen, wie es das DOS-PRESTO seit langem kann.

## Die Hauptfunktionen von *alcarta*

Mit [...] wird angedeutet, das das beschriebene Element eine Schaltfläche (engl. "Button") ist, siehe Abb. S. 2.

### [Index] (auch Alt+i)

Damit erscheint das Indexfenster, wo man zwischen bis zu 10 Registern umschalten und darin blättern kann. Wenn der Index geöffnet ist, kommt allgemeine Hilfe mit F1 und spezifische mit dem Button [?].

### [Find] (auch Alt+f)

Es öffnet sich ein Dialogfenster, in dem Suchbefehle erstellt werden können (s.Seite 14). Dieses Fenster hat ebenfalls seine eigene Hilfsfunktion. Auch eine Volltextsuche in der aktuellen Ergebnismenge kann veranlaßt werden.

### [←] / [→] (die Schaltflächen mit den Pfeilen, oben links)

Wie bei einem Web-Browser blättern diese Schaltflächen zwischen den vorher aufgesuchten Datensätzen rückwärts und vorwärts. Wenn Sie also einen Datensatz zurückholen wollen, der kurz zuvor auf dem Schirm stand: einfach ein paarmal [←] drücken. Oder die "History List" aufschlagen:

### Vorher angezeigte Daten

Das ist praktisch eine chronologische Übersicht der während der Sitzung aufgeblätterten Sätze. Die Entsprechung bei Netscape heißt "History List". Diese Übersicht kann beliebig lang werden. Mit der Taste [Entf] und dem Button [ENTF] kann man Einträge aus der Liste entfernen oder diese per [Besen] ganz beseitigen.

### Lesezeichen (engl.: Bookmark)

[Neues Lesezeichen] macht quasi ein Eselsohr an den gerade angezeigten Satz.

Mit [Zeige Lesezeichen] kommt die Übersicht der so markierten Sätze. Auch dies ist eine spezielle Ergebnismenge:

### Ergebnismengen

Die Liste im linken Fenster wird immer länger. Dort sammeln sich die Namen der angelegten Ergebnismengen. Wählt man eine aus, kommt die Kurzübersicht. Mit der [Entf]-Taste kann man Einträge in Ergebnismengen zum Löschen markieren und sie dann mit [ENTF] ganz daraus entfernen.

Wenn eine Ergebnismenge ausgewählt wird, erscheint außerdem der zugehörige ...

### Suchbefehl (auch Alt+s)

... unten im Eingabefeld. Dort kann man ihn modifizieren und sofort erneut ausführen lassen, um eine neue Ergebnismenge zu bilden. (F1 bringt spezielle Hilfe, wenn der Cursor in diesem Feld ist)

Hier ist sozusagen das Eingabefeld für die "Expertensuche", wie so etwas bei anderen Systemen mitunter genannt wird. Man muß dazu die *avanti*-Suchbefehle und die logischen Registernamen der Datenbank kennen (siehe Seite 14).

### [Besen]

Diese Schaltfläche ruft ein kleines Menü hervor, mit dem man die Ergebnismengen auf einen Schlag beseitigen kann, desgleichen die Lesezeichen und/oder die "vorher angezeigten Daten". Wenn das Programm als OPAC eingesetzt wird, kann damit jeder, bevor er das Gerät verläßt, seine Ergebnismengen komplett beseitigen, damit der nächste diese nicht "ausspähen" kann. Einzelne Ergebnismengen oder Einträge darin kann man auch mit der [Entf]-Taste beseitigen. Vor einem Export kann man somit unnötige Einträge von Hand herausnehmen.

### [Tür]

Hiermit wird ein externes Programm aufgerufen, das sich in der Datei DOOR.BAT verbirgt. Diese ist dem Anwender frei zugänglich, kann also individuell eingerichtet werden. Wenn dieses externe Programm verlassen wird, ist man wieder zurück in *alcarta*. Wenn das externe Programm eine Datei EXTERN.DAT produziert, die dem Format der Datenbank entspricht, kann diese Datei als "Externe Ergebnismenge" über das Menü "Datei" eingelesen, betrachtet, exportiert und im Volltext durchsucht werden. Dies ist auch der Anknüpfungspunkt für Z39.50-Recherchen in externen Datenbanken (s.a. S.10). Ansonsten lassen sich beliebige externe Programme auch über Flips starten, die z.B. eine zum Datensatz gehörige Grafik oder einen Volltext etc. anzeigen können.

### Aktuelle Ergebnismenge (auch Alt+b)

Die lange Schaltfläche unten in der Mitte trägt immer den Namen der aktiven Ergebnismenge, d.h. diese kann jederzeit sofort aufgeblättert werden. Das geht auch mit Alt+b (für "Browse"). Unten rechts sind noch zwei Buttons mit Pfeilen: damit blättert man in der aktuellen Ergebnismenge vor- und rückwärts (auch Alt+h / Alt+u).

Unter der Schaltfläche ist eine "Fortschrittsanzeige", die aktiv wird, wenn ein längerer Vorgang abläuft, z.B. ein Ergebnismengen-Export. In Windows-üblicher Weise wird dann der Fortgang der Aktion erkennbar.

### Menü

Die Funktionen auf dem Menü werden erklärt, wenn man die Hilfsseite unter '?' abrufen.

Eine datenbankspezifische Hilfe (die Liste der symbolischen Registernamen) erhält man mit dem Unterpunkt "Datenbank-Hilfe".

## Die wichtigsten Eigenschaften von *alcarta*

- Alle Einstellungen für eine Datenbank können in einer INI-Datei festgelegt werden (also nicht mit Aufruf-Optionen wie bei DOS-APAC). In der mitgelieferten Datei ALCARTA.INI stehen alle dabei möglichen Einstellbefehle mit ausführlichem Kommentar! Verwenden Sie diese Datei als Vorlage, wenn Sie eine Datenbank für *alcarta* benutzungsfertig machen wollen. Die Datenbank darf in jedem zugänglichen Verzeichnis des Dateisystems liegen, z.B. auch auf einer CD-ROM. Andererseits: Start mit `alcarta 1` ermöglicht Suche und Auswahl einer INI-Datei.
- Neu gegenüber *a99*: Das Anzeigefeld ist in RTF programmierbar. Darin kann markiert und Text über die Zwischenablage in andere Programme übernommen werden. ("Cut-and-Paste" mit Strg-a, Strg-c, Strg-v geht auch). In WinWord kommen alle Attribute herüber, auch die Farben und die Sonderzeichen des *allegro*-Zeichensatzes.
- "Hotlinks" oder "Hyperlinks" heißen bei *allegro* bekanntlich "Flips". Wenn man einen Flip doppelt anklickt, kann dahinter (a) ein Find-Befehl stecken (dann erscheint die Ergebnismenge bzw. ein anderer Datensatz), oder (b) ein Registersprung (dann erscheint der Registerabschnitt), aber auch (c) ein Programmaufruf. Das kann sogar Netscape sein, d.h. ein unmittelbarer Sprung aus *alcarta* ins Internet ist möglich (die CoOL-Datenbank demonstriert das). Außerdem kann bei *alcarta* ein Flip (d) eine andere Anzeigeform desselben Datensatzes auslösen. (e) Fünftens kann ein Flip auch eine Hilfsseite laden. (f) Und sechstens können auf Hilfsseiten wiederum Flips installiert sein (Beispiel: CATGER.RTF), d.h. man kann ein ausgedehntes, datenbankspezifisches Hilfesystem konstruieren. Flips können wie HTML-Links unterschiedlich aussehen, das ist Sache der Parametrierung. Meistens wird man sie als blau unterstrichene Textteile erscheinen lassen (wie oben zu sehen). Sie können auch in eckige Klammern eingeschlossen sein.
- Der Button mit dem Fragezeichen (unten links) bringt eine Hilfsseite hervor, die spezifisch für die Datenbank angelegt und dank RTF ansprechend gestaltet werden kann, z.B. CATGER.RTF für die Demo-Datenbank mit Namen CAT. Hilfetexte können wiederum Flips enthalten, die den Nutzer direkt zu wichtigen, interessanten oder instruktiven Stellen in den Registern hinführen. Die bereitgestellten Datenbanken haben alle eine derartige Hilfsseite. Hat eine Datenbank keine solche Seite, kommt eine allgemeine Hilfe.
- Das Programm ist auch zur Anbindung von Z39.50 Client-Funktionalität vorgesehen (DFG-Projekt), dafür ist die Schaltfläche mit der Tür gedacht ('W' wie Welt oder World oder Web). Das bedeutet, daß man im Anschluß an eine Abfrage dieselbe auch an andere Systeme weiterleiten kann, die über Z39.50 am Netz hängen. Die Ergebnisse können dann im *alcarta* so betrachtet werden, als seien sie Teil der eigenen Datenbank. (Konzept der "externen Ergebnismenge", siehe Beschreibung von *a99* in *allegro news* Nr. 51.) Dann stehen auch dem *allegro*-OPAC-Nutzer die einschlägigen Datenbanken in aller Welt offen. (In dieser ersten Version ist diese Funktion noch nicht enthalten.)
- *alcarta* ist ein monolithisches Programm, braucht also keinen Server. Daher ist es viel schneller als ein Client/Server-Programm überhaupt sein kann, praktisch so schnell wie DOS-APAC (obwohl es viel mehr macht). Manche erwünschte (und bei Web-Datenbanken oft vermißte) Details, besonders das zeilenweise, schnelle Blättern in beiden Richtungen in den Registern, können mit Client-/Server-Methodik gar nicht effizient realisiert werden.
- Die Fenster für die Register, die Kurzanzeige und die Suchbefehle sind die gleichen wie beim Datenbank-Editor *a99*, d.h. man hat auch die gesamte Mächtigkeit der *avanti*-Suchbefehle, die sortierbaren Ergebnismengen, Einschränkungen durch Erscheinungsjahr, Expansion ("Schiller-Räuber-Methode") etc.
- Auf dem Menü "Datei" findet man "DOS-APAC". Damit startet man das konventionelle Programm APAC, und zwar erscheint dann der Registerabschnitt, den man zuletzt geöffnet hatte. Wenn man eine oder mehrere Ergebnismengen exportiert (über das [Esc]-Menü), kann man dafür einen Namen vergeben. Bei Rückkehr nach *alcarta* hat man diese Daten als Ergebnismenge unter dem vergebenen Namen vor sich, kann aber damit alles machen, was *alcarta* kann.
- Über das Menü "Anzeige" können auch Listen und Kurzlisten von Ergebnismengen in die Anzeige gebracht werden, wo man sie beliebig markieren und dann in die Zwischenablage kopieren kann. Auf den Index- und Kurzanzeige-Fenstern sind ebenfalls Buttons zum Produzieren der Listen. Hierbei ist die Größe der Ergebnismenge auf etwa 64.000 Byte begrenzt. Größere kann man aber in eine Datei exportieren:
- Über "Export" läßt sich die Liste oder Kurzliste ebenfalls in eine RTF-Datei überführen, die man unmittelbar mit WinWord übernehmen kann. Hierbei ist die Größe der Ergebnismenge auf ca. 64.000 Titel begrenzt.
- Wie bei den anderen *allegro*-Programmen sind die Menütexte, Fehlermeldungen und die Beschriftungen der Buttons frei zugänglich (UIF-Datei), können also angepaßt werden. Mitgeliefert werden sie auf Deutsch und Englisch: UIFEGER und UIFEENG.
- Zum Programm gehören die zwei TTF-Schriften mit den bibliothekarischen Sonderzeichen (nach der Norm DIN 31628), die für das Windows-*allegro* entwickelt wurden. Diese Schriften können auch, wie jede TTF-Schrift, in den Windows-Standardprogrammen verwendet werden, insbesondere WinWord.

## Checkliste der *alcarta*-Dateien

Das nachfolgende Verzeichnis listet alle Dateien auf, die für *alcarta* notwendig oder nützlich sind. Angegeben ist neben Sinn und Zweck der Dateien auch, wo sie liegen müssen, sollten oder dürfen, und ob sie vom Anwender manipuliert werden müssen oder können. Studieren Sie die entsprechenden Dateien der Beispielpakete, diese bieten ausreichendes Anschauungsmaterial und enthalten nur diejenigen Dateien, die wirklich gebraucht werden.

Für die möglichen Standorte einer Datei verwenden wir folgende Bezeichnungen:

<b>PATH</b>	Die Verzeichnisse, die in der DOS-Variablen PATH stehen. Nur für Programme.
<b>DbDir</b>	Der Name des Datenverzeichnisses muß in der INI-Datei ausdrücklich angegeben sein (kann auch auf einer CD-ROM sein!). Solche Dateien sind datenbankspezifisch und gelten für alle Nutzer der Datenbank.
<b>ProgDir</b>	Default ist C:\ALLEGRO. Dort werden i.a. Dateien gesucht, die nicht auf DbDir und auch nicht lokal gefunden wurden. Solche Dateien gelten für alle Datenbanken, falls keine andere Version auf DbDir liegt.
<b>Lokal</b>	Startverzeichnis. Diese Dateien sind dann nutzerspezifisch. (Lokal geht immer, aber meist geht DbDir vor)

Welche Dateien muß oder kann man verändern? Konzentrieren Sie sich zuerst auf die mit [X] markierten.

Mit [X] sind die Dateien markiert, die datenbankspezifisch bearbeitet oder erstellt werden müssen. Die mit [o] sind optional, also nicht unbedingt notwendig. [v] bedeutet, daß die Datei verändert werden kann, aber nicht muß.

Name	Ort Verzeichnis-Reihenfolge, wo die Datei gesucht wird) Bedeutung, Funktion
------	--

-----

### 1. Allgemeine Dateien

(unabhängig von CFG und Datenbank, im allg. kein Eingriff nötig)

ALCARTA.EXE		Ort: PATH, Das Programm
ALC.BAT	[v]	Ort: PATH, Batch für Aufruf mit einer INI-Datei) <i>Voraussetzung:</i> DbDir XYZ hängt an Startverzeichnis, und XYZ.INI liegt auf XYZ. (Sonst evtl. ALC.BAT ändern.) Dann Start mit <b>alc xyz</b>
ALCARTA.INI		Kommentierte Liste aller INI-Befehle <i>Empfehlung:</i> als Vorlage für neue .INI verwenden
UIFOGER		Ort: Lokal / ProgDir. (nicht dbDir!) Default ist GER
UIFOENG		User-Interface. enth. u.a. die Button-Beschriftungen
UIFEGER	[v]	und Menütex te, wie fuer A99! (Auch Hotkeys veränderbar)
UIFEENG	[v]	GER = deutsche Version, ENG = englische Version
AS.BAT	[v]	Ort: PATH (Batch und Parameter fuer den APAC-Aufruf)
RS.Cpr	[v]	Ort: Lokal / ProgDir (Exportparameter für APAC) (wird automatisch als Kopie von RS.APR erzeugt bei Aufruf AS.BAT)
D-W0.APR		Ort: Lokal / ProgDir Default für SatzAnzeige, wenn D-W.cPR fehlt
E-W0.APR		Ort: Lokal / ProgDir Default fuer Export und externes Editieren wenn E-W.cPR fehlt
DOOR.BAT	[v]	Ort: PATH Aufruf eines externen Programms über die Schaltfläche mit der Tür. Wenn dieses eine Datei EXTERN.DAT erzeugt, kann diese anschließend eingelesen werden über das Menü "Datei" / "Externe Ergebnismenge" (s.a. Bericht Z39.50, S.10)

### 2. Datenbankspezifische Dateien dbn = Datenbankname (nur bis zu 4 Zeichen!)

*.INI	[X]	Name und Standort beliebig, denn Aufruf muß erfolgen mit <i>alcarta</i> #<Pfadname der INI-Datei>, also z.B. <i>alcarta</i> #c:\bolero\bol.ini oder <i>alcarta</i> #.\katalog.ini (wenn diese auf dem Startverz. liegt) <i>Empfehlung:</i> dbn.INI anlegen mit Standort DbDir. Wenn aber DbDir auf einer CD-ROM ist, wird die .INI-Datei anderswo liegen müssen. ALC.BAT dann anpassen!
c.CFG		Ort: DbDir / ProgDir, auch Lokal möglich [unveränd. DOS-Datei]
<DATENBANK>		Ort: DbDir Besteht aus dbn_*.cLD, dbn.cDX, dbn.TBL, dbn.STL, dbn.RES
Dbn.cPI	[X]	Ort: DbDir. [unverändert gegenüber DOS, bis auf:] <i>Wichtig:</i> Symbolische Registernamen einrichten! (➔ CAT.API)
D-WRTF.cPR	[X]	Ort: DbDir / ProgDir (aus D-1.cPR abgeleitet) (Vorgabe einer anderen mit Befehl OpacDisplay= in der .INI) Hier steckt in der Regel die schwierigste Aufgabe! Einbinden: D-RTF.APT (Attribute, Farbwerte) und D.APT
E-W.cPR		Ort: DbDir / ProgDir. Für Export und externes Edit. Wenn sie nicht da ist, wird die CFG-unabhängige E-W0.APR genommen, dieses

		macht einen ASCII-Export mit Kategorienummern. Vorgabe einer anderen mit Befehl ExportParameter= in .INI
P-W.cPR	[o]	Ort: DbDir / ProgDir. Fuer Export von formatierten Listen (z.B. Modifikation von D-WRTF.APR)
D-RTF.cPT	[v]	RTF-Steuerzeichen (Fett etc., Farben. Kopie von D-RTF.APT) (einbinden in Export- und Anzeigeparameter!)
????HEAD.RTF	[v]	Ort: DbDir / ProgDir / Lokal Vorspann-Dateien fuer RTF-Ausgabe. Werden automatisch benutzt, z.B. DISPHEAD wird vor die Satzanzeige gesetzt. Datenbankspezifische Versionen möglich (Schriftarten!) Lokale Anpassungen möglich (RTF-Kenntnisse)
HA_??GER	[v]	Allgemeine Hilfsdateien (→ Liste in HELP.TXT) Ort: Lokal / DbDir / ProgDir
HA???GER		(datenbankspezifische Versionen, andere Sprachen möglich)
dbnGER.RTF	[o]	Selbstgestaltete Hilfe zur Datenbank, mit WinWord erstellbar. Ort: DbDir dbn = Name der Datenbank. Vorbild: CATGER.RTF Default (wenn fehlend): HAGENGER
HA_InGER	[o]	Ort: DbDir n=1...9,: Registerspezifische Hilfsseiten, erscheinen bei Button [?] im Indexfenster Default wenn nicht vorhanden: HA_IXGER
*.RTF	[o]	Ort: DbDir / ProgDir (→ HELP.TXT) Spezifische, selbstgestaltete Hilfsdateien (Aufruf über Flips) Erstellung mit WinWord, Speicherung als .RTF

### 3. Verschiedene Hilfsdateien

O.APT	[v]	Ort: DbDir / ProgDir. / Lokal
D.APT	[v]	Tabellen für Zeichen-Umcodierung ASCII - ANSI Werden automatisch geladen, wenn Befehle to und td fehlen (.cPT-Versionen für eigene CFG anlegen, wenn nötig!) O.APT für Index- und Ergebnismengenanzeige D.APT nur für Satzanzeige- und -exportparameter
EBOX.ICO	[v]	Ort: egal. Wird als Icon für Desktop empfohlen
A-TIMES.TTF		<b>Müssen</b> ordnungsgemäß als Schriftarten installiert sein
A-LETTER.TTF		(Ort: meistens C:\WINDOWS\FONTS)

### 4. Dyamische Dateien (entstehen zur Laufzeit, kein Eingriff sinnvoll)

OUTPUT.ADT		Entsteht als Default Exportdatei (INI: OutputFile=...)
		Die folgenden Dateien existieren nur während einer Sitzung, oder wenn man die Sitzung später fortsetzen will (sonst bleiben nur die ersten zwei erhalten).
		Ort: immer Lokal (wichtig ist deshalb im Mehrbenutzer-Betrieb: jeder sollte sein eigenes Startverzeichnis haben)
DbName._1		Bookmarks
DbName._2		History list
DbName._i		Erg.Mengen (Nummernlisten) i=3,4,...
DbName.\$\$\$		Hilfsdatei mit Kopien von Sätzen
DbName.TAB		Adressentabelle für die Hilfsdatei
DbName.RSS		Liste der Ergebnismengen-Namen
DbName.RXX		Hilfstabelle (nur bei a99 wichtig)

## Die Z39.50-Schnittstelle für *allegro* (Cord Veltkamp)

Das von der Deutschen Forschungsgemeinschaft (DFG) seit zweieinhalb Jahren geförderte Projekt mit dem Titel "*allegro-Z39.50-Schnittstelle*" läuft zum Jahresende 1998 aus. Es ist somit an der Zeit, die Ergebnisse der Arbeit zusammengefaßt vorzustellen. Erfreulicherweise konnten die Projektziele vollständig erreicht werden. Verbesserungsmöglichkeiten bestehen noch auf dem Gebiet der Client-Funktionalität. Die dort noch laufenden Arbeiten betreffen nicht den Kern der Z39.50-Funktionalität, sondern die Integration in die parallel erschienenen, neuen Windows-Programme *alcarta* und *a99*. Mit Abstrichen am Komfort der Bedienung ist die Nutzung des *allegro-Z39.50*-Clients bereits jetzt möglich. In der nächsten oder übernächsten *news*-Ausgabe wird der Z39.50-Client ausführlich vorgestellt. In dieser Ausgabe liegt der Schwerpunkt auf dem *allegro*-Server-Modul, das es einem *allegro*-Datenbankbetreiber erlaubt, im weltweiten Netz als Datenanbieter aufzutreten, ohne das HTML-Protokoll zu benutzen.

Wer sich zunächst allgemein über das Z39.50 Protokoll informieren möchte, findet zahlreiche Links auf der Seite <http://www.ilrt.bris.ac.uk/discovery/z3950/resources/>

Die "offizielle" Dokumentation liegt bei der Library of Congress (<http://lcweb.loc.gov/z3950/agency/>).

Als Einstieg in das Thema empfiehlt sich der Vortrag von Bernd Hergeth "Z39.50 in Bibliotheken und im World-Wide-Web" von der 1. INETBIB-Tagung im März 1996 ([http://www.ub.uni-dortmund.de/Inetbib/v\\_herget.htm](http://www.ub.uni-dortmund.de/Inetbib/v_herget.htm)).

### Z39.50 - Was ist das?

Z39.50 ist die Nummer einer ANSI-Norm und steht für ein standardisiertes Kommunikationsprotokoll zwischen bibliothekarischen Datenbanksystemen (Server oder Target genannt) und den Zugriffsprogrammen (Client oder Origin). Z39.50 erlaubt die weltweite Suche in heterogenen Datenbanken aus der gewohnten lokalen Programmumgebung. Der Einsatz des Protokolls führt zu einer Unabhängigkeit von der Struktur der Datenbank (hierarchisch, relational, ...), der lokalen Abfragesyntax, dem eingesetzten Betriebssystem und der Hardware. Man kann sich das Z39.50-Protokoll als ein Datenbank-Esperanto vorstellen, das es jedem Client ermöglicht, mit jeder Datenbank einen Dialog zu führen (vgl. *allegro news* 44).

Wie bei jeder Standardisierung gibt es natürlich Kompromisse. Die Eigenheiten und Vorzüge des lokalen Systems finden im Regelwerk nicht immer eine Entsprechung. Zum Beispiel müssen das Rechercheprogramm und das Target-System sich auf ein Austauschformat einigen. International ist USMARC zumeist das Austauschformat der Wahl. Die Qualität der übertragenen Datensätze hängt dann von der Güte der Umwandlung vom Austausch- zum LokalfORMAT ab, in unserem Fall also von den Importparametern. Zum Glück können diese anwendungsseitig (ohne C-Programmierung) beliebig ausgebaut werden.

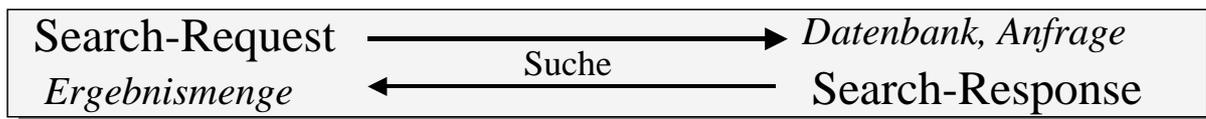
### Ein einfacher Z39.50 Dialog

Ein Client teilt dem Z39.50-Server den Kommunikationswunsch in einer Initialisierungsanfrage (Init-Request) mit:

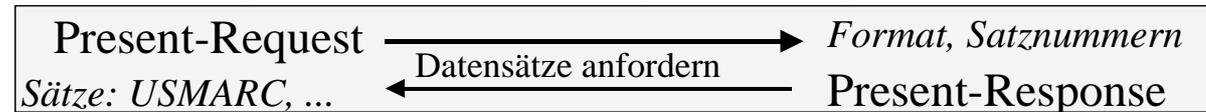


Der Client übergibt dabei einige Initialisierungsdaten, z.B. die Zugangskennung (Name, Passwort), die Versionsnummer des Z39.50-Protokolls, die Größe des lokalen Ergebnisspeichers. Der Server ist frei, die Verbindung anzunehmen oder abzulehnen. Steht einer Annahme nichts entgegen sendet der Server dem Client die akzeptierten Werte im Rahmen der Init-Response zurück und teilt ihm gleichzeitig mit, welche Protokolldienste er unterstützt.

Als ersten Dienst möchte der Client nun eine Suchanfrage (Search-Request) übergeben.



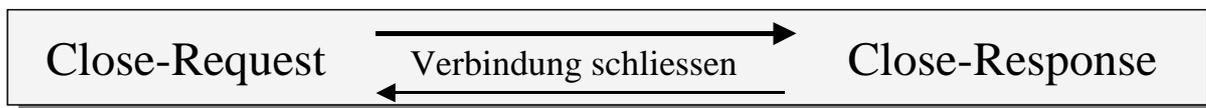
Dazu spezifiziert der Client die gewünschte Datenbank oder überläßt die Auswahl dem Server (Datenbankname „default“). Den Kern der Suchanfrage bilden natürlich die gewünschten Suchstrings, verknüpft mit den booleschen Operatoren AND, OR, NOT. Jeder Begriff kann durch Attribute differenziert werden. Die für bibliothekarische Anfragen entwickelte Attributsammlung BIB1 nennt 100 Kennzeichen, die den Suchbegriff erklären. Da die *allegro*-Datenbank normalerweise deutlich weniger Register aufweist, wird nicht jede Anfrage im gewünschten Sinn beantwortet werden können. Das ist jedoch normal: keine Datenbank bedient alle Attribute. Der Verwalter des Z39.50-Servers kann sich bei jedem nicht direkt unterstützten Attribut entscheiden, ob die Anfrage mit einer Fehlermeldung zurückgewiesen wird, oder ob die Suche auf ein verwandtes Register umgelenkt wird. Bei einer erfolgreichen Suche erhält der Client als Antwort die Zahl der gefundenen Datensätze. Der Server speichert die Ergebnismenge temporär. Auf Wunsch liefert der Server auch gleichzeitig ein paar Datensätze. Verbreiteter ist jedoch die Versendung der Sätze im Rahmen eines nachfolgenden Present-Requests.



Der Client übermittelt dabei den Startpunkt und die Anzahl der zu liefernden Datensätze aus der vorher erzeugten Ergebnismenge sowie das gewünschte Austauschformat. Der Server sendet als Antwort die formatierten Datensätze.

Die Kommunikation kann in dieser Weise beliebig fortgesetzt werden oder sich auf weitere Dienste ausdehnen. Im Unterschied zu Datenbankabfragen, die über das bekannte HTML-Protokoll abgewickelt werden, besteht bei einem Z39.50-Dialog eine permanente Verbindung zwischen Server und Client. Da der Server alte Ergebnisse bis zum Verbindungsabbruch speichert, kann der Client auf die Resultate zurückgreifen. Mit HTML ist so etwas viel schwieriger zu realisieren.

Ein Verbindungsabbruch wird protokollkonform mit einem Close-Request eingeleitet.



Als Antwort kommt ein Close-Response zurück und der Server löscht die zu der Verbindung gehörenden temporären Dateien. Dies passiert natürlich auch, wenn die Verbindung irregulär abbricht. Der Z39.50-Server kann die Verbindung zu jeder Zeit von sich aus schließen, zum Beispiel, um das System von untätigen Clients zu befreien.

**Was tut ein Z39.50-Server?**

Ein Z39.50-Server hat drei Hauptaufgaben:

1. Der Server muß die eintreffenden Z39.50-Anfragen dekodieren und in die lokale Datenbanksprache übersetzen. Umgekehrt werden die an den Client verschickten Antworten in die Z39.50-Sprache kodiert. Diese Aufgaben faßt man unter der **Protokoll-Funktion** zusammen.
2. Dank einer Auftragsverwaltung kann der Server die Aufträge der gleichzeitig aktiven Clients trennen. Meist soll nicht jedem Client jede Datenbank offenstehen. Die Verwaltung der Datenbanken (Zugriffskontrolle) gehört mit zur **Kontroll-Funktion** des Servers. Außerdem wird der Server in der Regel die Anfragen der Clients und seine eigene Reaktion protokollieren.
3. Die Kernfunktion eines Datenbank-Servers ist das Beantworten von Datenbankanfragen. Die Suche in der Datenbank und die Formatierung der gefundenen Sätze faßt man als **Datenbank-Funktion** zusammen.

Von dieser allgemeinen Gliederung weicht der *allegro*-Z39.50-Server nur in einem Punkt ab. Als ein externes Programm übernimmt der *avanti*-Server die Datenbank-Funktion und entlastet gleichzeitig das Target-Modul von der Verwaltung der Datenbanken. Z39.50-Target-Modul und *avanti*-Server bilden gemeinsam den *allegro* Z39.50-Server.



**Projektablauf**

Die Z39.50-Target-Entwicklung gliederte sich in drei Phasen, die in der folgenden Tabelle zusammengefaßt sind:

ab Juli 1996	Informationsrecherche <ul style="list-style-type: none"> <li>• Test und Auswahl der Z39.50 Funktionsbibliothek</li> <li>• Übersetzung der DBV-OSI II - Bibliothek unter Windows</li> </ul>
1997	<i>allegro</i> -Datenbankserver: <ul style="list-style-type: none"> <li>• Entwicklung des <i>avanti</i>-Servers für alle relevanten Plattformen.</li> <li>• Umstellung der WWW-Schnittstelle auf den <i>avanti</i>-Server: <i>allegro</i> kann nun auch als Client-Server-System fungieren und setzt sich als solches schnell durch.</li> </ul>
1998	Z39.50 Target-Funktionalität: <ul style="list-style-type: none"> <li>• Verbindung der DBV-OSI II-API mit dem Avanti-Server</li> <li>• Oktober 1998: Vorstellung auf dem <i>allegro</i>-Expertentreffen</li> <li>• Beginn der Beta-Phase, Freigabe mehrerer Datenbanken für Zugriffe von außen</li> <li>• Entwicklung eines ersten Client-Moduls in Verbindung mit dem neuen OPAC <i>alcarta</i></li> </ul>

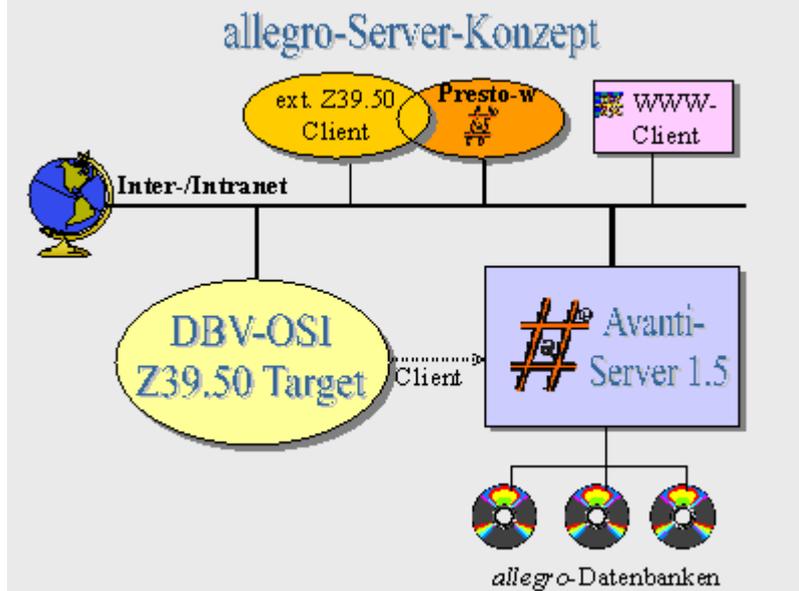
Während der Informationsphase wurde aus den im Internet verfügbaren Angeboten eine geeignete Z39.50-Funktionsbibliothek ausgewählt. Nach Prüfung der Alternativen fiel die Wahl auf das Softwarepaket der Deutschen Bibliothek, das im Rahmen des DBV OSI II - Projektes entstanden ist. Dieses Projekt hat sich die Verknüpfung der deutschen und europäischen Bibliotheksverbände zum Ziel und für die Kommunikation das Z39.50-Protokoll gewählt. Die Deutsche Bibliothek stellt die OSI-Software frei zur Verfügung. Da die Programme für UNIX-Rechner geschrieben sind, mußten sie zuerst unter Windows übersetzt werden. Die Anlehnung der *allegro*-Schnittstelle an einen großen Partner verspricht eine hohe Zukunftssicherheit der Entwicklung. Künftige Erweiterungen des Protokolls können mit geringem Aufwand übernommen werden.

In der zweiten Phase mußte *allegro* zu einem Client-Server-System ausgebaut werden. Der 1996 entwickelte *avanti*-Server erhielt eine TCP/IP-Schnittstelle und wurde in einer um zahlreiche Möglichkeiten erweiterten Version sowohl für UNIX als auch Windows den Anwendern übergeben. Zu dem Paket gehört eine WWW-Schnittstelle, die *avanti* als Client anspricht. Wie erhofft gibt es bereits von Anwendern erstellte Client-Programme. Sie zeigen, daß *allegro* sich als Client-Server-System durchsetzt. Obwohl der *avanti*-Server heute ein eigenständiges Programm ist, das viele Fähigkeiten besitzt, die vom Z39.50-Protokoll nicht genutzt werden, liegen seine Wurzeln im Z39.50-Projekt. Der *avanti*-Server übernimmt die Datenbankfunktion des Z39.50-Targets.

Die Kopplung der DBV-OSI-Bibliothek mit dem *avanti*-Server war die Aufgabe der dritten Projektphase. Auf dem *allegro*-Expertentreffen am 22./23. Oktober 1998 wurde der Z39.50-Target erstmals vorgestellt. Gleichzeitig wurden die Programme interessierten Anwendern zum Beta-Test zur Verfügung gestellt und mehrere Datenbanken zum Zugriff freigegeben.

## Das Zusammenspiel der Komponenten

Das enge Zusammenwirken des Z39.50-Moduls mit dem Avanti-Server verdeutlicht das folgende Blockdiagramm:



Der Z39.50-Target übersetzt die eintreffenden Anfragen in Aufträge an den Avanti-Server. Dieser arbeitet idealerweise auf demselben Rechner. Es ist aber auch eine physikalische Trennung der beiden Komponenten realisierbar und an der UB Braunschweig in Betrieb. Aus der Sicht des *avanti*-Servers ist der Z39.50 Target nur ein weiterer Client, von denen er bei Bedarf mehrere parallel bedient. Der *avanti*-Server übernimmt die Verwaltung der lokalen *allegro*-Datenbanken und liefert die Ergebnisse der Suche im gewählten Austauschformat an das Z39.50-Modul. Dort erfolgt die abschließende Codierung in die Z39.50-Sprache und die Übergabe an den externen Z39.50-Client.

Die Arbeitsteilung zwischen dem Z39.50-Target und dem Avanti-Server hat einige Vorteile:

- WWW-Zugriffe und Z39.50-Zugriffe laufen über denselben Datenbankservers, d.h.
- effiziente Entwicklung der Programme für WWW und Z39.50 (keine Doppelprogrammierung)
- zentrale Administration der Datenbanken (die Datenbanken sind den Clients unter einem alias-Namen bekannt).
- Stabilitätsgewinn durch die Trennung zwischen Z39.50-Target und Datenbankabfrage
- Datenbank und Z39.50-Target können auf getrennten Maschinen und Plattformen laufen

Der Nachteil einer verlängerten Antwortzeit fällt nicht ins Gewicht, da der limitierende Faktor der Kommunikation die Laufzeit im Netz ist.

Im Schaubild ist nicht eingezeichnet, daß auf die Datenbanken auch noch gleichzeitig mit den monolithischen Programmen PRESTO, *a99* und *alcarta* zugegriffen werden kann. Diese brauchen weder den *avanti*-Server noch das Internet/Intranet, sondern nur die Berechtigung, die Dateien der Datenbank im Mehrplatzmodus zu öffnen. Dies ist auf allen Plattformen einschließlich Novell und NT möglich. Auf eine *entfernte* (also nicht lokale) Datenbank werden in Zukunft aber *a99* und *alcarta* ebenfalls mit Hilfe von Z39.50 als Clients zugreifen können.

**TIP zum Einbinden eines Fremd-Client-Programms:** *alcarta* übergibt an das Programm DOOR.BAT (Schaltfläche "Tür") mehrere Parameter, als ersten den zuletzt gegebenen Suchbefehl. Wenn man einen bestimmten Z39.50-Client benutzt, kann man in DOOR.BAT diesen starten, unter Verwendung des Suchbefehls. Recherche-Ergebnisse können, immer noch in DOOR.BAT, mittels IMPORT in das eigene Format umgewandelt und als EXTERN.DAT bereitgestellt werden, damit *alcarta* diese anschließend einlesen kann (über das Menü "Datei" / "Externe Ergebnismenge").

## Z39.50-Dienste

Die DBV-OSI Funktionsbibliothek befindet sich auf dem Stand V3 des Z39.50-Protokolls. Wie die meisten Z39.50-Server unterstützt auch der *allegro*-Target nicht alle Dienste des Protokolls. Die mit einem Haken gekennzeichneten Funktionen sind zur Zeit verfügbar. Der Client kann einen Registerauszug anfordern (Scan), Begriffe in der Datenbank suchen (Search) und sich formatierte Datensätze schicken lassen (Present). Die Suchbegriffe können mit alten Ergebnismengen logisch kombiniert werden. Werden einzelne Ergebnismengen nicht mehr benötigt, können sie gezielt gelöscht werden (Result-Set-Delete). Uns ist kein Standard-Client bekannt, der weitergehende Protokollfunktionen nachfragt. Die Entwicklung des *allegro*-Targets wird natürlich fortgeführt. Als zusätzliche Dienste sollen im nächsten Jahr der Extended Service, Sort und Resource-Control hinzukommen. Andere Dienste, insbesondere der Explain-Service, müssen wegen der größeren Komplexität und der geringeren praktischen Bedeutung noch länger auf die Implementierung warten.

- Initialise ✓
- Search ✓
- Present ✓
- Result-Set Delete ✓
- Scan ✓
- Close ✓
- Sort >
- Resource Control >
- Extended Service >
- Resource Report »
- Segment »
- Explain »

## Plattformen

Der *allegro*-Z39.50-Server ist für alle Betriebssysteme verfügbar, auf denen auch der *avanti*-Server läuft. Das sind die folgenden Plattformen:

UNIX	Windows
Linux - Kernel 2.x	Windows NT 4.0
Sun-Solaris 2.5	(Systemdienst)
Dec-Alpha 4.0b	
HP-UX 9.05	
IBM-AIX	

Windows 95/98 wird nicht unterstützt, da sich dort kein Systemdienst aufsetzen läßt und die Stabilität des Betriebssystems Defizite aufweist. Windows NT jedoch ist als Plattform gut geeignet und hat inzwischen eine sehr hohe Bedeutung. Unter UNIX wird der Server normalerweise als Hintergrundprozeß, Daemon genannt, gestartet und auf einem Windows NT-Rechner entsprechend als Systemdienst. Die notwendigen Schritte sind in der Installationsanleitung beschrieben.

## Installation des *allegro*-Z3950-Servers

Im ersten Installationsschritt muß der aktuelle *avanti*-Server installiert werden. Der *avanti*-Server wird mit einer Demodatenbank ausgeliefert, die auch als Testumgebung für den Z39.50-Target dienen sollte, da sich die beigefügten Parameterdateien stets auf das A-Schema beziehen. Nachdem die Funktion des *avanti*-Servers überprüft ist, kann die Installation des Z39.50-Targets beginnen.

Das Z39.50-Serverprogramm hat keine Benutzeroberfläche. Die notwendigen Einstellungen werden in Initialisierungsdateien nach dem Muster einer Windows-INI-Datei eingetragen.

Zwei Typen von Steuerdateien sind zu unterscheiden:

### a) zentrale Einstellungen

Die zentralen Einstellungen, die für jede neue Verbindung von Bedeutung sind, werden in die Datei **.target\_z39\_ini** (UNIX) bzw. **target\_z39.ini** (NT) eingetragen. Hier legen Sie z.B. die Portnummer des Servers fest und bestimmen den Umfang der Protokollfunktion.

### b) datenbankspezifische Vorgaben

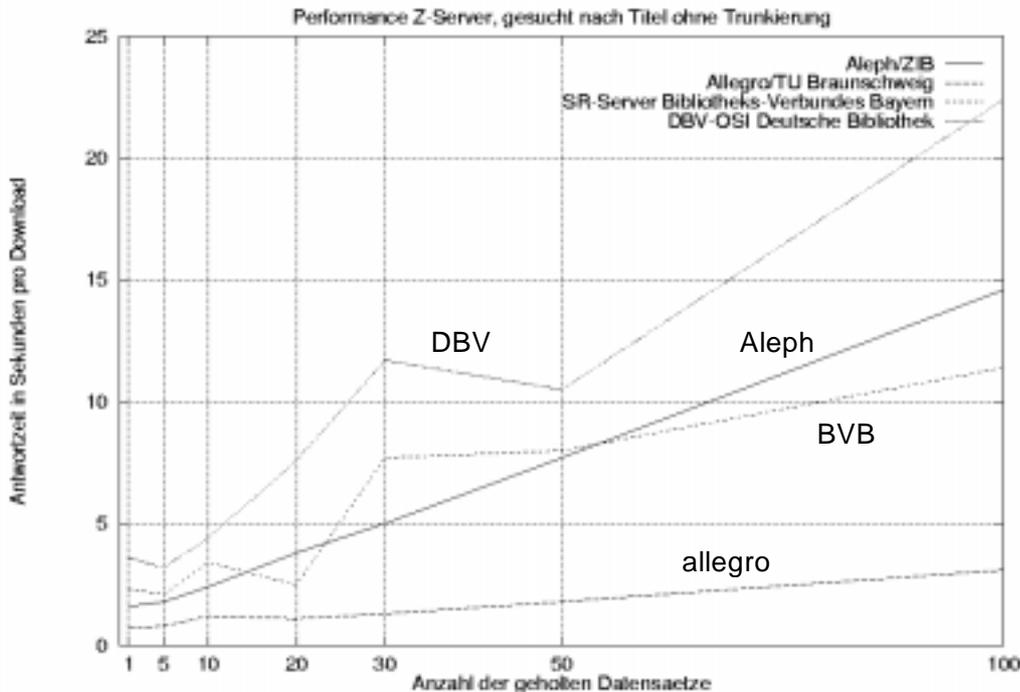
Die zweite Gruppe sind INI-Dateien mit datenbankspezifischen Angaben, welche die Verbindung zwischen der Z39.50-Anfrage und dem lokalen *avanti*-Auftrag herstellen. Hier werden z.B. die Z39.50 Attribute auf die *allegro*-Register umgelenkt und die Export-Parameterdateien der Austauschformate festgelegt. Anhand der auf die Demodatenbank ausgelegten, kommentierten Beispiele ist es sehr einfach, die Steuerdatei für die eigene Datenbank anzulegen. Mehr dazu steht in der Installationsanleitung.

Während das Erstellen einer Steuerdatei mühelos von der Hand geht, ist es nicht immer einfach, die Export-Parameterdateien für das Austauschformat zu schreiben. Für das Standard-Schema A.CFG liefern wir im Unterverzeichnis AVDEMO Vorlagen für die Formate USMARC, MAB2 und UNIMARC (letztere erstellt von Herrn Allers, München). Falls eine Datenbank ein anderes Schema verwendet, ist u.U. in diesen Teil der Installation einige Arbeit zu investieren. Unter den *allegro*-Experten gibt es aber auch einige Anwender, die solche Aufgaben gegen Entgelt erledigen, etwa als Werkvertrag.

## Betatest

Seit die Software auf dem Expertentreffen 98 erstmals der *allegro*-Öffentlichkeit präsentiert wurde, haben sich zahlreiche Anwender am Beta-Test beteiligt. Die zurückfließenden Anregungen und Fehlermeldungen resultierten umgehend in verbesserten Versionen. Unser besonderer Dank gilt dabei der Projektgruppe „Kooperativer Bibliotheksverbund Berlin-

Brandenburg“ im Konrad-Zuse-Zentrum für Informationstechnik Berlin. Dort plant man den Zusammenschluß heterogener Bibliotheken mit Hilfe des Z39.50-Protokolls und hat die *allegro*-Software mit anderen Programmen verglichen. Ein Teilaspekt war dabei die Antwortgeschwindigkeit. Aufgrund des effizienten Aufbaus einer *allegro*-Datenbank kann der Braunschweiger Server in dieser Kategorie Pluspunkte sammeln.



Quelle: Wolfram Schneider: Ein verteiltes Bibliotheksinformationssystem auf Basis von Z39.50. Diplomarbeit an der TU Berlin

Bei der Anzahl der zur Verfügung stehenden Dienste schneiden andere Server zum Teil besser ab. Entsprechend den tatsächlichen Anforderungen der Anwender wird der Funktionsumfang des *allegro*-Target-Moduls aber ausgebaut werden.

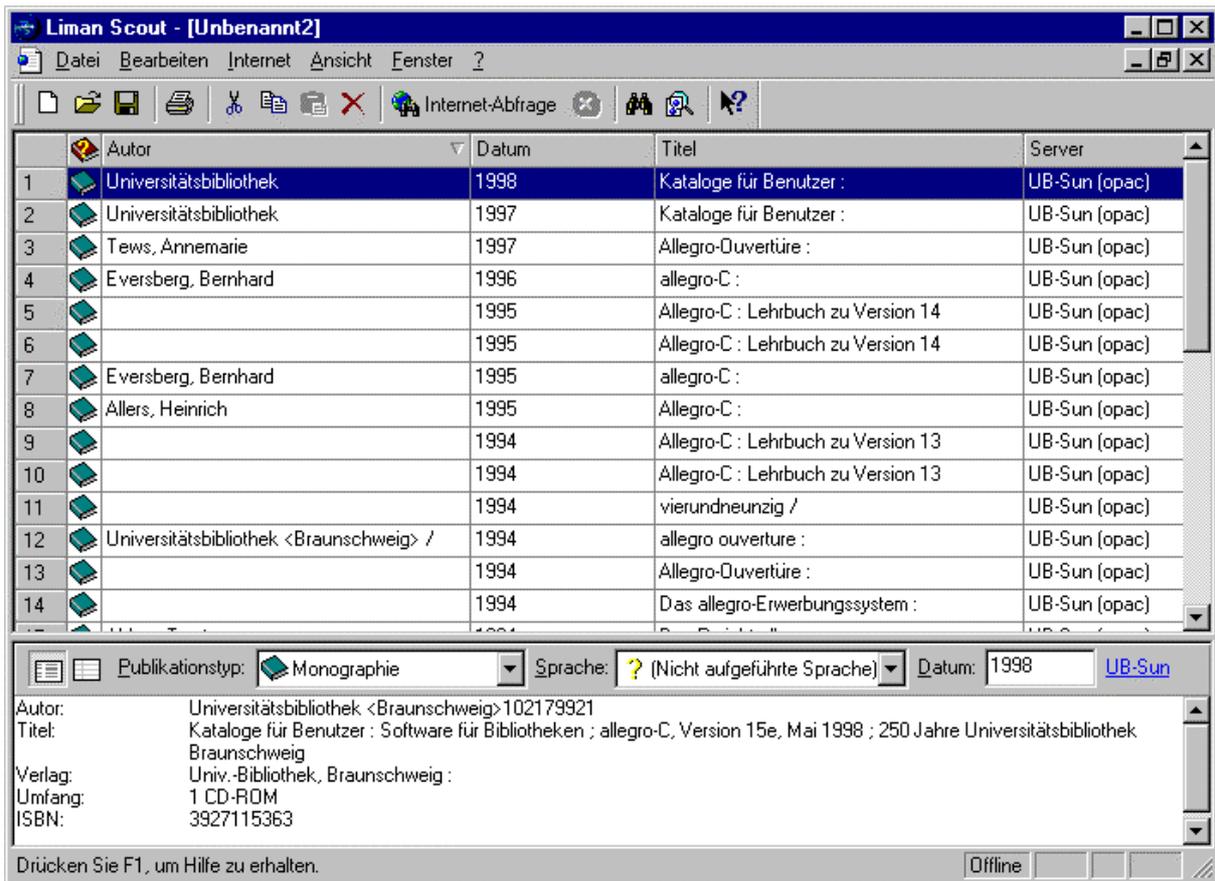
Wenn Sie sich für das *allegro*-Z39.50-Modul interessieren, sind Sie eingeladen, sich am Beta-Test zu beteiligen:

1. Der erfaßte Lokalbestand der UB-Braunschweig (jetzt 580.000 Sätze) ist seit Oktober mit Hilfe des *allegro*-Z39.50-Servers recherchierbar. Als Recherchewerkzeug benötigt man ein beliebiges Z39.50-fähiges Client-Programm. Als Einstiegsprogramme empfehlen sich der frei verfügbare *ZNavigator* (<http://www.sbu.ac.uk/~litc/caselib/software.html>) oder als ein deutschsprachiges Programm der günstige *Liman-Scout* (<http://www.liman.de/>). Unter Wissenschaftlern beliebt sind außerdem die Literaturverwaltungsprogramme *EndNote* (<http://www.endnote.com/>) und *BookWhere* (<http://www.bookwhere.com/>). Die kommerziellen Anbieter stellen eine in der Laufzeit begrenzte Testversion ihrer Programme kostenlos zur Verfügung.

Um mit dem Braunschweiger Z39.50-Server in Kontakt zu treten, geben Sie in die Client-Software die folgenden Verbindungsdaten ein:

IP-Adresse	ubsun01.biblio.etc.tu-bs.de
Port	2020
Datenbank	Opac
User	Opac
Password	Opac

Die folgende Abbildung zeigt die Ergebnisse einer Suche mit dem *Liman-Scout*. Der Suchbegriff war „**allegro**“ und das BIB1-Attribut 4=Titel. Die in USMARC erhaltenen Ergebnisse werden vom *Liman-Scout* in einer sortierten Liste angezeigt. Im Unterschied zu vielen internationalen Programmen setzt der *Liman-Scout* deutsche Umlaute korrekt um.



Liman-Scout: Suche nach dem Stichwort „allegro“

2. Wenn Sie selbst einen Z39.50-Server installieren möchten, können Sie sich die beschriebene Software von unserem FTP-Server herunterladen. Sie finden das Verzeichnis **Z3950** neben dem *avanti*-Server für die gewünschte Plattform. Nach dem Entpacken erklärt die Installationsanleitung „install.htm“ alle notwendigen Schritte. Fehlermeldungen und Verbesserungsvorschläge senden Sie bitte per E-Mail an die *allegro*-Liste [allegro@buch.biblio.etc.tu-bs.de](mailto:allegro@buch.biblio.etc.tu-bs.de).

Nach Ablauf der Beta-Phase ist geplant, auf der *allegro*-Homepage eine Z39.50-Anwender-Seite aufzulegen. *allegro*-Anwender, die ihre Datenbanken über das Z39.50-Protokoll zur Recherche freigeben möchten, können sich dann dort eintragen und Profile ihrer Datenbanken bereitstellen. Eine solche Fundgrube hat sich bereits bei der WWW-Schnittstelle bewährt : [www.allegro-c.de/allegro/ac-dbs.htm](http://www.allegro-c.de/allegro/ac-dbs.htm)

## Zum Jahr-2000-Problem

Es wurde mehrfach schon erwähnt, daß die *allegro*-Programme bis auf *aLF* immer schon die Jahreszahlen vierstellig behandelt haben und das gefürchtete Problem daher kein Thema ist - jetzt auch nicht mehr für *aLF* (die Anwender wurden rechtzeitig informiert und mit einer gründlich geprüften, neuen Version beliefert, die im übrigen auf dem Niveau V15e ist.).

Ein Punkt bleibt noch - keine Fehlfunktion, aber eine Ungenauigkeit: Die Eingabepfung der Kategorie #76 hatte bisher immer festgestellt, ob die Jahreszahl größer als 1449 und kleiner als 2000 war. Wenn nicht, kam ein Hinweis - keine Fehlermeldung, denn die Kategorie wurde trotzdem akzeptiert. Diese Unebenheit wäre aber wohl zu einem Ärgernis geworden, daher wurden die Programme (PRESTO, MENUED, ALFA etc.) verbessert: sie prüfen jetzt, ob die Jahreszahl nicht größer ist als das laufende Jahr +2. Die CD wird diese Programme enthalten, per FTP sind sie schon zugänglich auf AC15/PROG.

Keinen Einfluß hat jedoch die Entwicklungsabteilung darauf, was Anwender in ihren Parametern machen! Da kann es durchaus vorkommen, daß Jahreszahlen nur 2stellig gespeichert, indiziert oder bei Berechnungen nur die letzten zwei Ziffern benutzt werden. Wir empfehlen daher dringend, sich die eigenen Parameter daraufhin anzusehen. Wir haben selbst die Indexparameter CAT.API geprüft und mehrere Stellen gefunden, wo die '1' des Erscheinungsjahres (in #76), eine Rolle spielte, z.B. in Manipulationsbefehlen wie **b"1"** oder **b"19"**. So etwas funktioniert nicht mehr, sobald man "2000" in der #76 hat. Wieder so ein Fall, wo das Programm nicht mitdenken kann. Wir haben an solchen Stellen andere Konstruktionen eingebaut, z.B. so etwas wie **#76 x"\*1" e4 p"|6"** anstatt **#76 b"1" e3 p"|61"**.

Nach diesem Muster können Sie sich auch bei der Überprüfung Ihrer Parameter richten: suchen Sie in den Parameterdateien, vor allem in Indexparametern, ob dort Zeichenkombinationen wie **b"1"** oder **p"1"** vorkommen. Prüfen Sie aber auch, ob Jahreszahlen irgendwo 2stellig gespeichert sind. Dann sind Globale Ersetzungen oder Manipulationen fällig.

Die korrigierte CAT.API liegt auf dem Verzeichnis PARAM des FTP-Servers und wird auf der kommenden CD-ROM mitgeliefert. In den zu *aLF* bzw. *ORDER* gehörigen Varianten sind diese Stellen gleichfalls bereinigt.

**avanti-Suchbefehle** (für die *Expertensuche*, d.h. Befehlseingabe von Hand im Eingabefeld "Suchbefehl")

Der *avanti*-Server hat eine Befehlssprache, die u.a. den Befehl **find** enthält. Dieser ermöglicht logische Kombinationen und Klammerungen. Dieselbe Befehlslogik ist auch in *alcarta* eingebaut. Vor langer Zeit, in der Ausgabe 40 (über das Programm VP), haben wir dargestellt, wie solche Befehle aussehen können. Ansonsten steht es in der *avanti*-Dokumentation. Beides hat nicht jeder bei der Hand, daher soll diese Seite das Wesentliche knapp zusammenfassen.

Das Befehlswort **find** entfällt bei *alcarta*, ansonsten ist die Syntax aber dieselbe. Durch einen Befehl entsteht eine Ergebnismenge. Evtl. besteht sie nur aus einem Satz, dann wird dieser sofort angezeigt, in der Ergebnismengenliste wird dann aber kein Eintrag gemacht. Die maximale Größe der Erg.Menge ist vorerst 64.000.

Für die logischen Kombinationen gibt es die Befehlswörter **AND, OR, NOT**. Verschachtelte Klammerung ist möglich.

*Beispiel:* ('?' ist das Trunkierungssymbol)

```
per beethoven? and (tit klavier? or tit piano?)
```

Es werden Sätze mit "beethoven" als Person und "klavier" oder "piano" als Titelstichwort gesucht, jeweils trunziert.

*So geht's aber auch:*

```
|1 beethoven? and ( |3 klavier? or |3 piano?)
```

Diese Form funktioniert immer, ohne Eingriff in die Indexparameter. Man muß dafür jedoch die Registernummern wissen. Die weitaus bequemeren symbolischen Befehle wie `per goethe?` kann man nur verwenden, wenn für die betreffende Datenbank **symbolische Registernamen**, wie eben **per** für das Register der Personen, definiert sind. Welche Registernamen eine Datenbank hat, erfährt man über den Menüpunkt "?": Datenbank-Hilfe.

Der Verwalter des *allegro*-Systems kann Registerbezeichnungen jederzeit einführen oder ändern (ohne Neu-Indexierung!).

Dafür müssen in der Index-Parameterdatei (z.B. CAT.API) Befehle dieser Art stehen:

```
I XXX r "Klartextbezeichnung des Registers"
      XXX ist ein symbolischer Registername
      r ist die Nummer des Registers (1..9, :, ;)
```

Die Klartextbezeichnungen fehlen in älteren CAT.API-Versionen noch!

z.B.:

```
I PER 1 "Personenregister"
I AUT 1 "Personenregister"
I TIT 3 "Titelstichwörter"
```

Man erkennt: ein Register kann mehrere Namen haben, die in den Befehlen gleichwertig sind (hier: PER = AUT).

Sonderabteilungen in Registern können eine eigene Bezeichnung bekommen:

```
I DIS 1D. "Dissertationen unter Ortsnamen"
```

Bei CAT.API gibt es im Register 1 die Sonderabteilung D mit Leerzeichen hinter dem D (deshalb der Punkt): dort sind die Dissertationen unter Ortsname,Jahr aufgelistet.

Mit dieser Definition kann man z.B. so suchen:

```
dis berlin,1992? //finde alle Berliner Dissertationen des Jahres 1992
```

Wenn Restriktionen definiert sind (siehe *news* 46) können diese in Suchbefehlen wie folgt verwendet werden:

```
(normaler Suchbefehl) and ERJ >1990
```

Dazu muß eine R-Zeile in den Indexparametern existieren, die z.B. so aussieht:

```
R ERJ r1 "Erscheinungsjahr"
```

Das bedeutet, daß man **erj** als Restriktionsbefehl verwenden kann.

Sollen bei einer logischen Kombination zwei Begriffe aus demselben Register kombiniert werden, braucht der Registername bei *alcarta* nur beim ersten angegeben zu werden, also z.B.

```
per beethoven? or mozart? ist gleichwertig mit per beethoven? or per mozart?
```

Sind schließlich für die Datenbank sog. SR-Schlüssel eingerichtet, kann mit "Expansion" gesucht werden:

```
per &schiller? and tit räuber um das bekannte "Schiller-Räuber-Problem" zu lösen (s. news 46).
```

### E-Mail-Diskussionsliste

Man schreibt sich in die *allegro*-Liste ein, indem man an die Adresse `maiser@buch.biblio.etc.tu-bs.de` eine Botschaft mit nur einer Zeile sendet: `subscribe allegro`. Man wird dann sofort eingetragen und erhält eine Mitteilung mit weiteren Informationen, insbesondere, wie man sich wieder abmeldet. Die Liste hat ca. 250 Teilnehmer.

... Frühere Ausgaben, aktuelle Programmversionen etc.: **FTP:** 134.169.20.1 oder **WWW:** [www.allegro-c.de](http://www.allegro-c.de)