

# Berichte

aus der Biologischen Bundesanstalt für Land- und Forstwirtschaft

## Reports

from the Federal Biological Research Centre for Agriculture and Forestry

---

Heft 119

2003

### Workshop Datenmanagement

Workshop Data Management

herausgegeben von  
edited by

Eckard Moll\*) und Thomas Stauber\*\*)

\*) Biologische Bundesanstalt für Land- und Forstwirtschaft  
Zentrale EDV-Gruppe

\*\*) Bundesforschungsanstalt für Forst- und Holzwirtschaft  
Institut für Forstgenetik und Forstpflanzenzüchtung



**BBA**

---

Herausgeber

Biologische Bundesanstalt für Land- und Forstwirtschaft  
Braunschweig, Deutschland

## Vorwort

Vom Gesamtaufwand entfallen 80 % und mehr auf die Datenaufbereitung und in der Regel weniger als 20 % auf die Datenanalyse und Datamining. Deshalb lagen Interesse und Notwendigkeit beisammen, sich mit Problemen und Erfahrungen des Datenmanagements auseinander zu setzen. Teilnehmer aus dem Geschäftsbereich trafen sich am 30. April 2002 zu einem Workshop.

Foren im Sinne des Erfahrungsaustauschs sind im Rahmen der Fortbildungskurse des Senats der Bundesforschungsanstalten im Geschäftsbereich des Bundesministeriums für Verbraucherschutz, Ernährung und Landwirtschaft eher die Ausnahme. Um so mehr sagen die Herausgeber den Initiatoren des *Workshops Datenmanagement*, den Akteuren und den Teilnehmern für ihre rege Diskussion Dank.

Die Vortragenden erklärten sich bereit, aus ihren PowerPoint-Präsentationen Artikel zu fertigen. Das dauerte leider etwas länger, so dass die nunmehr schriftlich vorliegenden Beiträge mit etwas Verzug in einem dem Workshop gewidmetem Heft zusammengefasst vorliegen.

Nach 2 mehr grundlegenden Artikeln zur Datenverwaltung werden in Nutzung befindliche Lösungen vorgestellt. Diese realisierten Aufgaben zeigen unterschiedliche Komplexität und Mann-Power. Sie eint das Ziel, eine effiziente, transparente und zuverlässige Datenverwaltung als Grundlage verschiedener Zusammenfassungen, Recherchen und Auswertungen zu schaffen.

Für den Inhalt der Manuskripte sind die Autoren verantwortlich.

Die Herausgeber danken der Biologischen Bundesanstalt für Land- und Forstwirtschaft, die Beiträge in einem separaten Heft zum *Workshop Datenmanagement* zusammen zu fassen.

Eckard Moll  
Thomas Stauber

**Verlag**

Eigenverlag

**Vertrieb**

Saphir Verlag, Gutsstraße 15, D-38551 Ribbesbüttel

Telefon +49(0) 5374 6576

Telefax +49(0) 5374 6577

**ISSN** 0947-8809

**Kontaktadresse**

Dr. Eckard Moll

Biologische Bundesanstalt für Land- und Forstwirtschaft

Zentrale EDV-Gruppe, Außenstelle Kleinmachnow

Stahnsdorfer Damm 81

D-14532 Kleinmachnow

Telefon +49(0) 33203 48 331

Telefax +49(0) 33203 48 424

E-Mail [E.Moll@bba.de](mailto:E.Moll@bba.de)

Internet <http://www.bba.de>

© Biologische Bundesanstalt für Land- und Forstwirtschaft

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersendung, des Nachdrucks, des Vortrages, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf fotomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.

## Inhaltsverzeichnis

Versuch einer Begriffsbestimmung zum Thema „Datenmanagement“ (Worseck, S.)	5
Grundlagen datenbasierter Informationssysteme (Klöß, D. und Schröder, R.)	7
Datenmanagement am Beispiel des <u>Informationssystems</u> für die <u>Zulassung</u> von <u>Pflanzenschutzmittel</u> – INFOZUPF (Worseck, S.)	17
TSN – bundeseinheitliches Werkzeug zum Krisenmanagement von Tierseuchen (Schröder, R., Klöß, D., Kroschewski, K., Staubach, C. und Micklich, A.)	35
Datenbankmanagement mit raumbezogenen Inhalten (Staubach, C., Klöß, D., Schröder, R. und Busse, W.)	43
Datenmanagement in der BAZ am Beispiel der Kohl-Datenbank (Kecke, S.)	49
Anwesenheitsliste	63

# Versuch einer Begriffsbestimmung zum Thema „Datenmanagement“

Worseck, S.

Biologische Bundesanstalt für Land- und Forstwirtschaft

(s.worseck@bba.de)

Datenmanagement bedeutet mehr als nur das Handling von Daten. Im Sinne von Management = Leitung ist der Begriff auf alle Bereiche auszudehnen, die das Umfeld des Datenhandlings maßgeblich bestimmen.

## 1 Einleitung

Einem Nicht-Informatiker steht es nicht unbedingt zu, über die Begriffswelt der Informatiker zu urteilen. Meinem Empfinden nach werden im Sprachgebrauch der Informatiker immer wieder neue schillernde Begriffe geboren, wie z.B. Datenmanagement, Datenanalyse, Künstliche Intelligenz, Business Intelligence, Informationsmanagement, Data Warehousing, Customer-Relationship-Management, e-Business, ... die jedoch mit unterschiedlichen Halbwertszeiten wieder verblassen.

MERETZ (2002) formulierte meinen Eindruck vom Standpunkt der „kritischen Informatiker“ :

*„ In der Informatik ist die Situation fatal und besonders schwierig. Fatal ist, dass die Informatik kein Fundament hat, auf dem sie aufbaut. Eine Wissenschaft zeichnet sich durch die Grundbegriffe aus, die sie konstituiert und von denen sie ausgeht. Auch in der Informatik gibt es Versuche, solche Grundbegriffe zu definieren. Pech ist nur, dass es zu einem Grundbegriff, wie etwa der ‚Information‘, mindestens 20 verschiedene Definitionen gibt. Es gibt also eigentlich kein einheitliches Verständnis, was man da eigentlich tut, nur womit man es tut, darüber ist man sich einig: mit dem Computer. Doch ein Gerät konstituiert keine Wissenschaft! Schwierig wird die unklare Situation der Informatik, weil es scheinbar gar nicht notwendig ist, Klarheit zu schaffen. Die Informatik ‚funktioniert‘ auch so. ... Gleichzeitig ist die Rede von der Krise der Informatik als Krise der Softwareentwicklung schon bald 30 Jahre alt. Sie kehrt genauso immer wieder wie die Verkündigung der endgültigen Bewältigung der Krise. Das alles macht eine kritische Beschäftigung mit dem Thema schwierig“.*

Ohne eine entsprechende Begriffswelt ist die Beschäftigung mit dem Thema Datenmanagement nicht möglich. Zu einem Klassiker soll das Buch“ Unternehmensweites Datenmanagement - von der Datenbankadministration bis zum modernen Informationsmanagement“ von DIPPOLD u.a. (2001) gehören.

## 2 Datenhandling – Datenmanagement

Aus diesem Grund sollen die Begriffe Datenhandling und Datenmanagement aus eigener Sicht etwas näher untersetzt werden.

Mit Daten kann man arbeiten (**Datenhandling**). Man kann sie:

- erheben,
- speichern, sortieren, filtern, archivieren,
- vor Datenverlust und
- unberechtigtem Zugriff schützen,
- sie ständig dem aktuellen Stand in ihrem Wert anpassen,
- sie über Schnittstellen austauschen,
- über verschiedenste Protokolle weiterleiten,
- Datenformate wechseln,
- sie von "Datenmüll" trennen,
- in den Daten Inkonsistenzen und Datenfehler finden und korrigieren,
- separat oder in Verbindung mit anderen Daten auswerten,

• sowie in Abhängigkeit der Datenauswertungen Aktionen bzw. Schlussfolgerungen daraus ableiten.  
Sobald diese Daten aber

- für Schlüsselfunktionen in einem Unternehmen notwendig sind, oder
- es sich Massendaten handelt oder
- sich die Vorgänge ständig unter Zeitdruck wiederholen,
- oder unter Einbeziehung vieler Schnittstellen,
- bzw. vieler Kooperationspartner gearbeitet wird
- ...

ist ein **Datenmanagement** notwendig. Im Sinne von Management = Leitung ist der Begriff auf alle Bereiche auszudehnen, die das Umfeld des Datenhandlings maßgeblich bestimmen.

Dafür sind

- die notwendigen inhaltlichen, organisatorischen, personellen, materiellen und finanziellen Rahmenbedingungen zu schaffen und
- es müssen die entsprechenden Werkzeuge und Methoden zum richtigen Zeitpunkt vorhanden sein.

Vielfach müssen Richtungsentscheidungen getroffen werden, ohne dass man das Gesamtprojekt überhaupt vollständig durchdringen konnte. Es kommt darauf an, die Tragweite von Richtungsentscheidungen

bei der Wahl von äußeren Rahmenbedingungen z.B.

- Datenbanksystem,
- Hard- und Softwarearchitektur,
- Entwicklerwerkzeuge,
- Kauf der Hardwarebasis

bei der Wahl von inneren Rahmenbedingungen z.B.

- von Kodierungssystemen,
- Ausnutzung bestimmter Schutzrechtsmechanismen,
- Programmiervorgaben

beim Wechsel einer der Rahmenbedingungen z.B.

- Softwareupdates (Betriebssystem Server/Client, Entwicklungs-Tools, Datenbanksystem

zum **richtigen Zeitpunkt** in **vollem Umfang** zu erkennen.

## Literatur

MERETZ S. (2002): Was ist kritische Informatik?  
<http://www.kritische-informatik.de>

DIPPOLD R., MEIER A., RINGGENBERG A. u.a. (2001): Unternehmensweites Datenmanagement. Von der Datenbankadministration bis zum modernen Informationsmanagement  
Vieweg Verlag (ISBN 3528256613)

# Grundlagen datenbasierter Informationssysteme

**Klöß, D. und Schröder, R.**

**Bundforschungsanstalt für Viruskrankheiten der Tiere**

**(detlef.kloess@wus.bfav.de , ronald.schroeder@wus.bfav.de)**

Ziel des Artikels ist es, einen Datenbanklaien mit einigen Aspekten der Entwicklung von Informationssystemen auf der Grundlage von Datenbanken vertraut zu machen.

Häufig werden umfangreiche Informationen mit Systemen wie Excel oder gar auf Papier als so genannte Fragebögen erfasst. Wenngleich dieses Vorgehen zunächst zweckmäßig erscheint, z.B. weil sofort mit der Datenerhebung begonnen werden kann, entstehen Probleme oft dann, wenn die Daten ausgewertet werden sollen oder der Datenumfang mit Papierbögen und Excel-Tabellen nicht mehr zu bewältigen ist.

Der Beitrag soll anhand eines leicht nachvollziehbaren Beispiels zeigen, wie eine Datenbank grundsätzlich aufzubauen ist. Dabei wird herausgestellt, dass bereits beim Entwurf eine konkrete Abgrenzung der Datenbasis erfolgt, d.h., es werden genau die Informationen gespeichert, die zur Erfüllung einer bestimmten Aufgabe erforderlich sind, es werden Wertebereiche und Schlüsselssysteme festgelegt und definiert, welche Angaben obligatorisch und welche optional sind. Einige Grundbegriffe des relationalen Datenbankmodells werden dabei erläutert und gezeigt, wie Tabellen über gemeinsame Schlüsselfelder miteinander verknüpft werden.

## 1 Ziele

Anhand von leichtverständlichen Beispielen soll in diesem Beitrag gezeigt werden, wie man Daten speichern und auf diese zugreifen kann. Dazu werden einige theoretische Grundlagen beleuchtet und maßgebende Begriffe der Datenbanktheorie erläutert. Besonderes Augenmerk wird darauf gelegt, dass die Sicht des Endanwenders für die Probleme der Entwickler geschärft werden soll.

## 2 Erfahrungen

Im Rahmen der vielfältigen Programmierarbeiten in einer veterinärmedizinischen Forschungseinrichtung entstehen oftmals neue Projekte durch folgende oder ähnliche Aufgabendefinitionen:

*„Wir haben hier 500 Antworten auf unseren Fragebogen und es kommen noch mehr. Schreibt uns dafür mal ein Programm. Aber es muss erweiterbar sein und flexible Auswertungen ermöglichen ...“*

Zwischen den Erwartungen des Anwenders und den Anforderungen des Entwicklers an die Aufgabenstellung bestehen oftmals erhebliche Diskrepanzen. Im Einzelnen sind das:

Anwender erwartet	Entwickler erwartet
<ul style="list-style-type: none"> <li>• Schnelle Ergebnisse</li> <li>• Einfache Bedienung</li> <li>• Erweiterbarkeit</li> <li>• Flexibilität</li> </ul>	<ul style="list-style-type: none"> <li>• Klare Formulierung bzw. Abgrenzung des Problems</li> <li>• (durchdachte) Datenmodelle</li> <li>• Paradigma der Programmentwicklung <b>Ausgabe – Eingabe – Verarbeitung</b></li> </ul>

Als Vergleich können auch einige Analogien herangezogen werden:

- Wenn ein Kuchen gebacken werden soll, müssen alle Zutaten vorher eingekauft werden oder
- Wenn eine Party gefeiert werden soll, sollte man die Gäste im Vorfeld befragen, was sie trinken wollen.

In den meisten Fällen sind zeitliche Vorstellungen für die Fertigstellung nicht vorhanden bzw. sind völlig unreal.

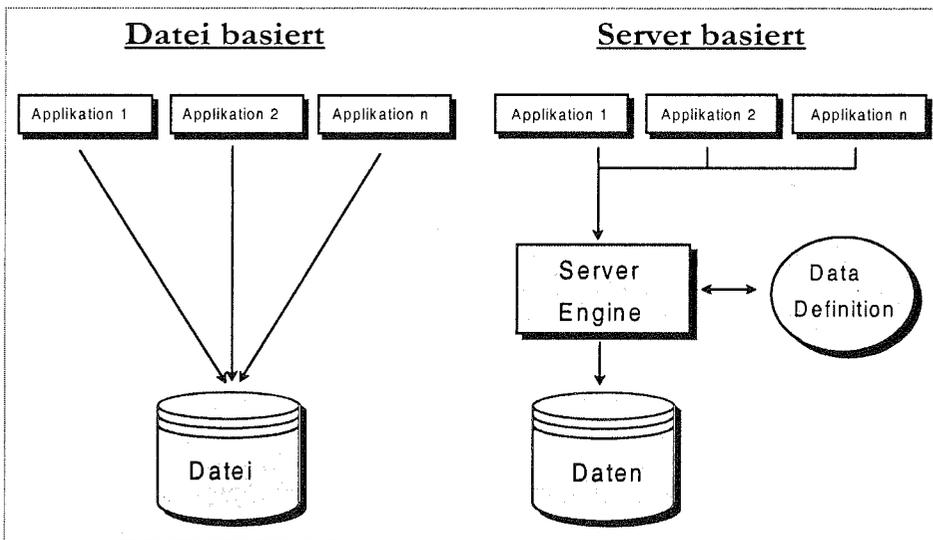
### 3 Wie kann man Daten speichern?

Um Sachdaten elektronisch abzulegen, gibt es eine Vielzahl von Möglichkeiten. In der folgenden Tabelle werden drei gegenübergestellt:

Textdateien	Arbeitsblätter	(relationale) Datenbanken
<ul style="list-style-type: none"> <li>• Unstrukturiert</li> <li>• Nur Volltextsuche</li> <li>• Automatisierte Auswertungen kaum möglich</li> </ul>	<ul style="list-style-type: none"> <li>• Einfache, flexible Auswertung für eine Tabelle</li> <li>• Starke Mengenbegrenzung (64.000 Zeilen, 255 Spalten)</li> <li>• Kaum Fehlertolerant (die einzelnen Daten-spalten sind nicht so normiert, dass nur gleichartige Inhalte eingetragen werden können)</li> </ul>	<ul style="list-style-type: none"> <li>• Effiziente Datenspeicherung</li> <li>• Flexibler Zugriff</li> <li>• Sicherheitsmechanismen</li> <li>• Beinahe beliebiger Umfang der Daten</li> </ul>
z.B.: Word-, HTML-Dateien	z.B.: Excel-Tabellen	z.B.: SQL-Datenbanken

In letzter Zeit gewinnt das XML-Format zunehmend an Bedeutung. XML (eXtensible Markup Language) ist eine Strukturbeschreibungssprache für logische Auszeichnungen und zugleich ein nicht proprietäres, freies Datenformat, das den Datenaustausch zwischen verschiedenen Programmen und Plattformen ermöglicht. Alle XML-Dokumente bestehen aus reinem ASCII-Text. Um XML-Dokumente interpretieren zu können werden sogenannte Parser benötigt. Das sind Programme, die die syntaktische XML-Struktur erkennen und in andere Formate (z.B. Datenbankformate) überführen können.

In der Art und Weise, wie auf die Daten zugegriffen wird, unterscheidet man zwei wesentliche Strategien. Zum einen handelt es sich um den Datei basierten Zugriff. Dem steht der Server basierte (Client-Server) gegenüber. Die folgende Abb. stellt die beiden Mechanismen gegenüber.



Der wesentliche Unterschied besteht darin, dass bei der Server basierten Methode eine leistungsstarke Software (Server Engine) alle Anforderungen an den Datenbestand durchführt. Diese steht praktisch zwischen den Anwendungsprogrammen (Applikationen) und den Daten. Bei der Datei basierten Zugriffsart muss diese Logik in die Applikation integriert werden.

## 4 Anwendungsprogramme und Datenbanken

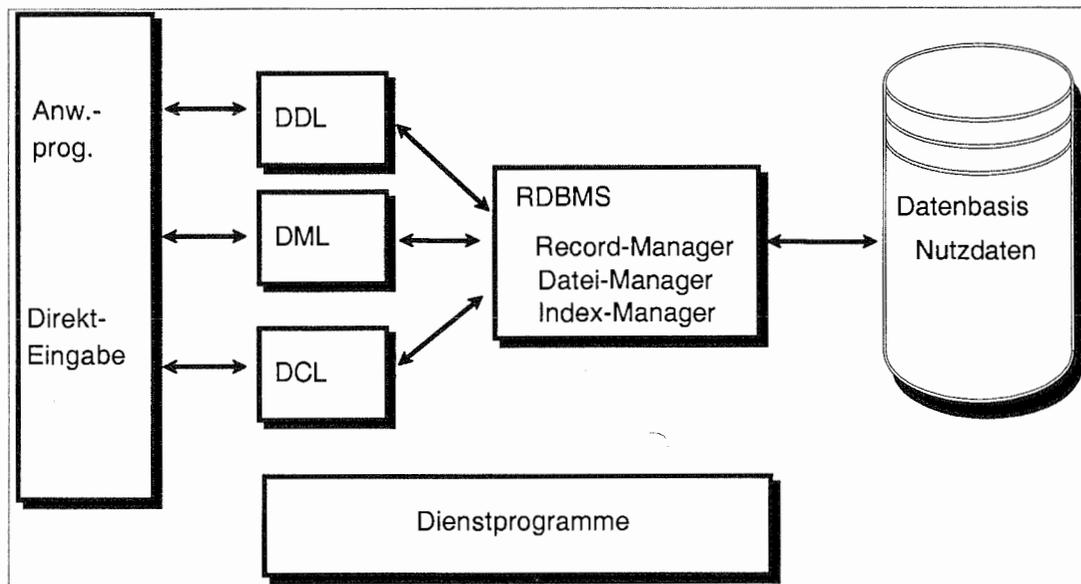
### Datenbank (eine Definition)

- Selbständig, auf Dauer und flexiblen Gebrauch angelegte Datenorganisation
- Umfasst Datenbestand und die dazugehörige Datenverwaltung

### Datenbank (Software-Komponenten)

- Datenbankverwaltungssystem (DBMS)
- Datendefinitions- u. Datenmanipulationssprachen (DDL, DML)
- Zugriffskontrolle (DCL)
- Dienstprogramme (z.B. Import, Export)
- Systemtabellen (z.B. Datenkatalog, Zugriffstabellen)

Die folgende Abb. zeigt das Zusammenwirken von Anwendungsprogrammen, Software-Komponenten und Datenbasis:



Die Abb. zeigt eine typische Client-Server-Architektur. Alle notwendigen Komponenten zur Datenmanipulation und -abfrage sind i.d.R. in der Datenbank vorhanden, es erfordert aber Spezialwissen (z.B. SQL), um die Datenbank zu bedienen. Des Weiteren sind die Schnittstellen der Datenbank aus Anwendersicht „unhandlich“. Aus diesen Gründen wird eine Trennung von Daten (Datenbank, Backend) und deren Verwendung (Anwendungsprogramme, Frontend) vorgenommen.

Somit kann man die Fähigkeiten der verwendeten Programmiersprache zur Realisierung anspruchsvoller grafischer Benutzeroberflächen nutzen und effiziente Mechanismen der Datenbank zur Datenhaltung verwenden.

## 5 (Einige) Grundlagen relationaler Datenbanken

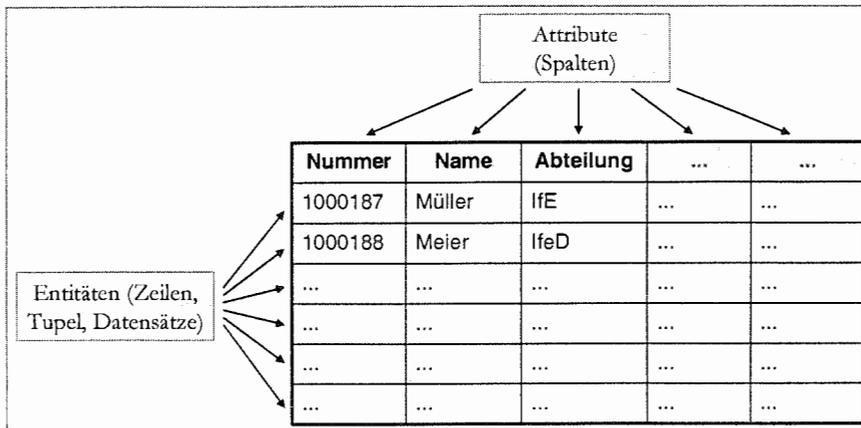
Im Folgenden werden einige grundlegende Begriffe bzw. Grundlagen relationaler Datenbanken definiert. Eine mögliche Definition des Begriffs „Datenbank“ lautet:

„ ... Datenbank, dass ist die Verknüpfung von Zeilen beliebig vieler Tabellen über gemeinsame Schlüssel-felder ...“

Weitere Begriffsdefinitionen:

**Relation (=Tabelle)**

- ... Zusammenfassung von Entitäten mit gleichen Eigenschaften
- **Entitäten (=Tupel, Zeilen, Datensätze)**
  - ... Identifizierbares Exemplar von Dingen, Personen oder Begriffen aus der realen oder Vorstellungswelt
- **Attribute (=Spalten)**
  - ... Beschreibung einer bestimmten Eigenschaft (von Entitäten einer Entitätsmenge (Entitätsmenge = Entitäten mit gleichen Eigenschaften) )
  - Eigenschaften haben Wertebereiche (Domäne)
    - Statisch: z.B. Geschlecht (männlich, weiblich)
    - Dynamisch: z.B. Kundennummer



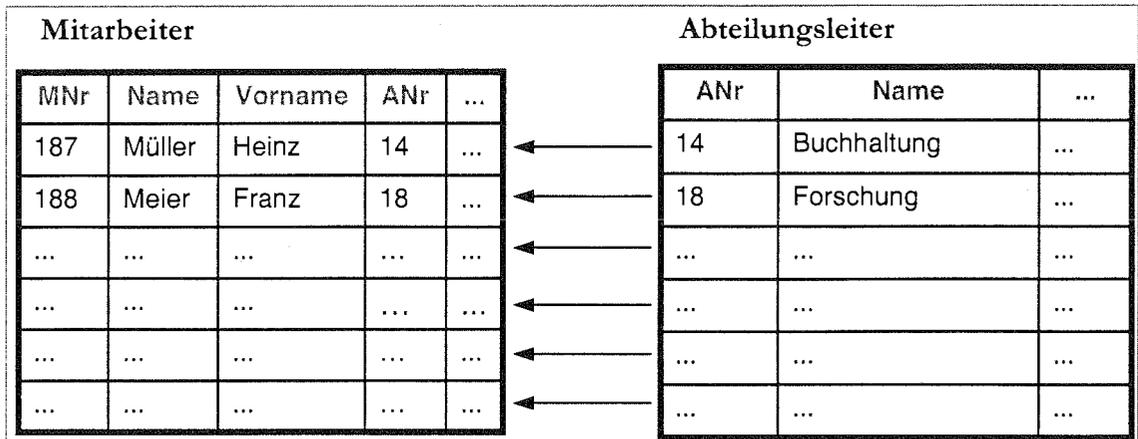
**Schlüssel**

- Merkmal, mit dem Elemente (Entitäten, Datensätze) in einer Menge von Elementen ausgezeichnet werden können
- Schlüsselarten
  - Identifikationsschlüssel (Primärschlüssel)
    - identifizieren eine Entität (Zeile, Datensatz) in einer Entitätsmenge (Tabelle) eindeutig
      - Eindeutigkeit (einmalig, unveränderlich)
      - Sofortige Zuteilung (neue Identität erhält Schlüssel sofort)
      - Kürze, Schreibbarkeit
  - Suchschlüssel
  - Sortierschlüssel

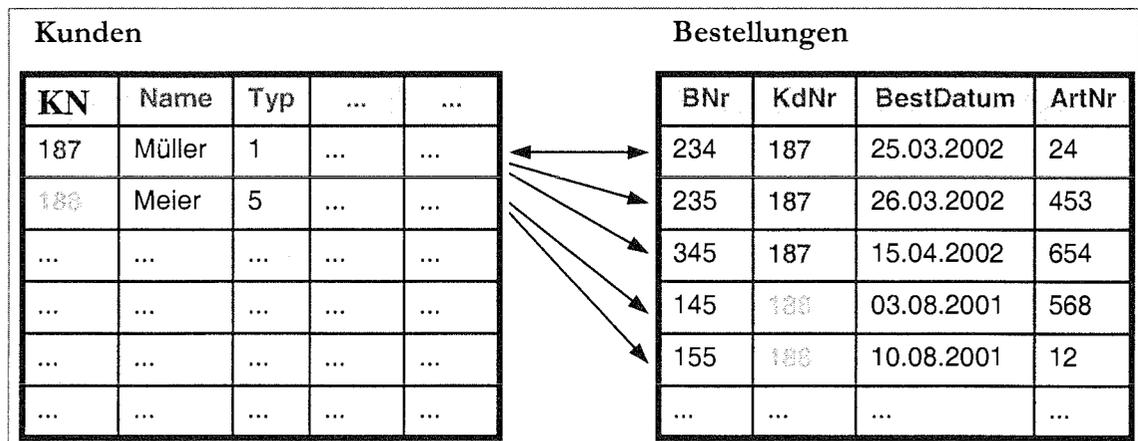
**Beziehungen**

- Verbindung von zwei oder mehrer Entitäten (Zeilen) miteinander
- Primärschlüssel – Fremdschlüssel
 

Jede Zeile einer Tabelle besitzt einen eindeutigen Identifikationsschlüssel (Primärschlüssel, Primary Key). Um Tabellen miteinander zu verknüpfen, wird dieser Primärschlüssel in die zu verknüpfende Tabelle als so genannter Sekundärschlüssel (Fremdschlüssel, Foreign Key) einbezogen.
- Häufige Verknüpfungen
  - 1-1 :



o 1-n :



**Normalformen**

- ... werden gebildet, um Redundanzen in einer Datenbank zu beseitigen
- Redundanz ist vorhanden, wenn ein Teil der Daten ohne Informationsverlust weggelassen werden kann

Die nächste Abb. veranschaulicht das Prinzip der Normalisierung:

Nicht Normalisiert

Bestellungen						
BestNr	ArtNr	ArtName	ArtPreis	KNr	KName	Bonität
101	401	Bohrmaschine	180	3412	Bauehof Müller	1
102	525	Drehbank	1.500	3412	Bauehof Müller	1
103	401	Bohrmaschine	180	2111	Autohaus Meier	2
104	401	Bohrmaschine	180	2533	Elektromeister Blitz	4

Normalisiert

Bestellungen			Artikel		Kunden		
BestNr	ArtNr	KNr	ArtNr	ArtName	KNr	KName	Bonität
101	401	2111	401	Bohrmaschine	2111	Autohaus Meier	2
102	525	2433	525	Drehbank	2533	Elektromeister Blitz	4
103	401	2111			2111	Autohaus Meier	2
104	401	2433			2533	Elektromeister Blitz	4

Die obere Tabelle stellt eine nicht normalisierte Tabelle dar. Die Redundanzen (Wiederholungen) sind farbig markiert. Neben dem erhöhten Speicherbedarf besteht ein sehr hoher Änderungsaufwand, sobald einzelne Datenwerte geändert werden müssen. Soll z.B. der Artikelname der „Bohrmaschine“ auf „Schlagbohrmaschine“ verändert werden, muss der entsprechende Wert drei Mal geändert werden. Die unteren drei Tabellen stellen denselben Sachverhalt in normalisierter Form dar. Die Tabelle „Bestellungen“ beinhaltet nun nur noch die Bestellnummern, die Artikelnummern und den Kundennummer. Alle beschreibenden Daten stehen in den Tabellen „Artikel“ und „Kunden“. Muss nun die Bezeichnung der Bohrmaschine geändert werden, bracht dieser nur in der Tabelle „Artikel“ einmalig verändert zu werden.

**Integrität**

- **Datenkonsistenz**
  - Logische Richtigkeit der Daten, Widerspruchsfreiheit (Datenmanipulation nur mit Transaktionen möglich)
- **Datensicherheit**
  - Technische und organisatorische Maßnahmen zur Schutz vor physischem Datenverlust, Beschädigung und unerlaubtem Zugriff auf den Datenbestand

*Beispiele:*

- Informationen zu obligatorischen Felder müssen immer vorhanden sein
- Wenn Angaben zu obligatorischen Feldern nicht verfügbar sind, dürfen keine willkürlichen Werte erfasst werden, um den Datensatz abspeichern zu können
- „Weniger ist manchmal mehr“, d.h., lieber wenige Datenfelder genau erfassen, als viele Felder vorsehen und nur wenige wirklich mit Daten füllen

**Datenbank-Operationen**

- Abfragen (Query, SQL) - Abgrenzung eines bestimmten Inhaltes der Datenbank
- Datenmanipulation - beliebige Operationen auf der Datenbank
  - Mutation - Veränderung des Inhaltes eines bestimmten Bereiches der Datenbank
- Transaktion - Konsistenz erhaltende Operation auf der Datenbank; Zusammenfassung von Teiloperationen auf Datenbeständen, die nur gemeinsam oder gar nicht ausgeführt werden dürfen (z.B. wird ein Eintrag aus der Tabelle „Artikel“ gelöscht, müssen alle eventuell in der Tabelle „Bestellung“ vorhandenen Einträge für diesen Artikel gelöscht werden)

**6 Datenmodell am Beispiel TSN**

Das *TierSeuchenNachrichten*-System (TSN) ist das System zur Computer gestützten Erfassung, Auswertung und Übertragung von Informationen zu amtlich festgestellten anzeigepflichtigen Tierseuchen und meldepflichtigen Tierkrankheiten der Bundesrepublik Deutschland. Über dieses System werden die zuständigen Veterinärbehörden tagesaktuell über das nationale Tierseuchen- und -krankheitsgeschehen informiert. SCHRÖDER u.a. (2003) nehmen eine umfassende Beschreibung des TNT vor.

## Informationsumfang

Zu einem Seuchenobjekt (damit ist das Gehöft gemeint, in dem eine Tierseuche bzw. Tierkrankheit auftrat; bei Wildtieren wird so der Seuchen- bzw. Krankheitsfall bezeichnet) werden die folgenden Informationen gespeichert:

- Kreis
- Kennnummer Seuchenobjekt (Typ-Tierseuche, Jahr, lfd. Nummer.)
- Gemeinde
- Ortsteil
- Tierhalter (nur für die Kreisveterinärbehörde)
- Koordinaten des Seuchenobjektes
- Tierseuche
- Virustyp/Erreger
- Diagnoseverfahren
- Datum Seuchenfeststellung
- Datum Seuchenaufhebung
- Untersuchungsgrund
- Angaben zu empfänglichen Tieren (Haltungsstufen):
  - bei Seuchenobjekt-Anmeldung
    - Tierart           ▪ im Bestand
    - erkrankt       ▪ verendet
    - getötet         ▪ geschlachtet
    - Bestandsart
  - bei Nachmeldungen
    - Datum           ▪ Tierart
    - Zu-/Umsetzungen
    - erkrankt       ▪ verendet
    - getötet         ▪ geschlachtet
- Erkrankter Bestand vor Seuchenfeststellung gegen Seuche geimpft
  - Datum letzter Impfung
- Einschleppungsweg:
  - gesichert
  - vermutet
- Datum Seuchenverdacht
  - seuchenverdächtige Tierart
  - Anzahl
- vermuteter Tag der Erstansteckung
- getroffene Maßnahmen
  - Datum
  - Anzahl
  - Tierart
- Datum Abschluss Tötung
- Datum Abschluss Schlachtung
- Entfernung zum Nachbarbetrieb
- Zusammenhang zu Seuchenobjekt
- Bemerkungen

Weiterhin werden Daten zu Tierhaltungen und sonstigen Betrieben gespeichert.

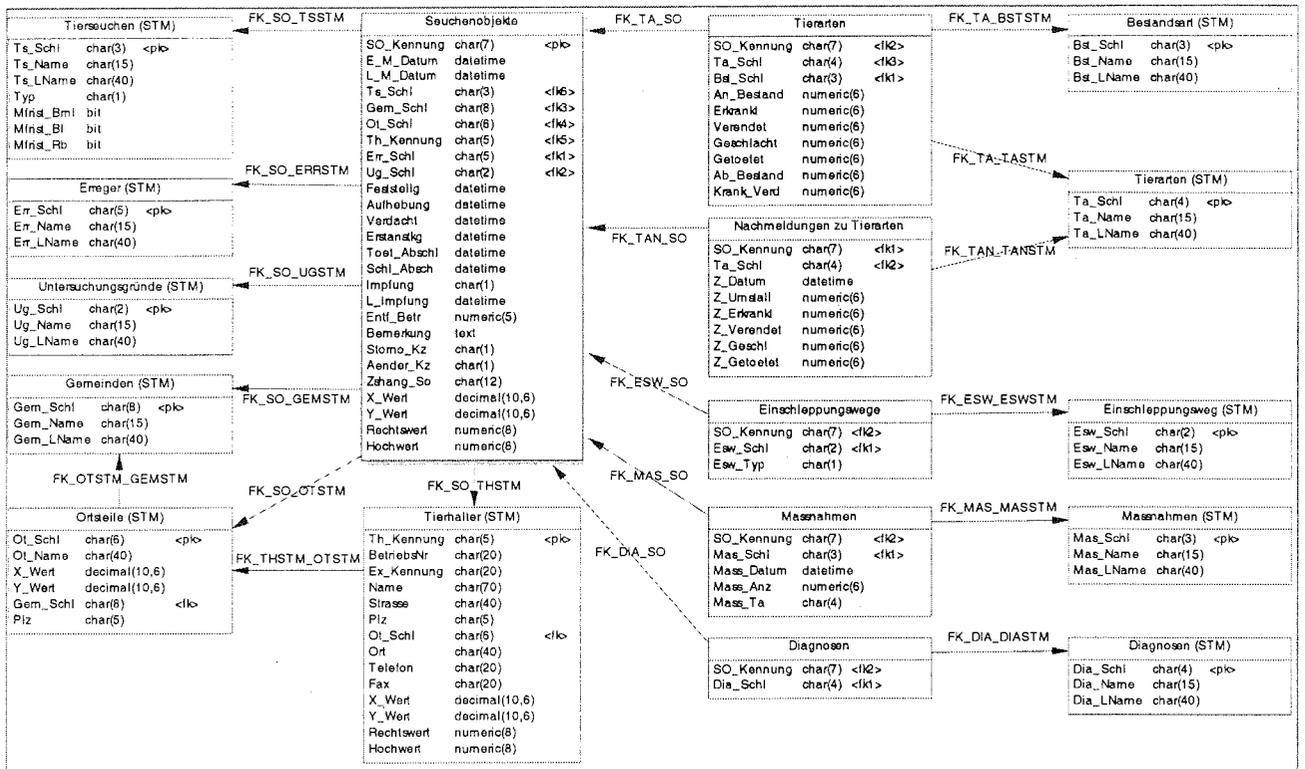
### Tierhaltung

- Betriebsnummer
- Adresse
- Koordinaten
- durchschnittliche Bestandszahlen für ausgewählte Tierartkategorien
  - Nutztiere
  - Geflügel/Fische
  - Sonstige/Wild

### sonstige Betriebe

- Betriebsnummer
- Adresse
- Koordinaten

Aus dem Informationsumfang wurde das nachfolgend dargestellte Datenmodell entwickelt:



Jeder Kasten stellt eine Tabelle der Datenbank dar. Die grau hinterlegten Tabellen beinhalten die Bewegungsdaten. Die anderen, durch „STM“ gekennzeichneten Tabellen, stellen so genannte Stammdatentabellen dar. Dies sind Steuertabellen, die zentral vorgegebenen und gepflegt werden. In den Tabellen sind die einzelnen Datenfelder mit ihren Datentypen und Längenangaben aufgeführt. Die Abkürzungen „<pk>“ und „<fk>“ markieren Datenfelder, die in dieser Tabelle Primär- (<pk> - Primary Key) bzw. Fremdschlüssel (<fk> - Foreign Key) sind.

## 7 Zusammenfassung

Im Rahmen der vorliegenden Abhandlung werden einige Grundlagen dargestellt, die im Zusammenhang mit Datenbasierten Informationssystemen stehen. Dabei wird besonderen Wert darauf gelegt, dass der Endnutzer bei der Definition seiner Anforderungen für die Probleme der Systementwickler (Programmierer) sensibilisiert wird.

Ausgehend von der Speicherung der Daten in

- Textdateien,
- Arbeitsblättern,
- (relationalen) Datenbanken oder
- XML-Dateien .

werden dann die Datei basierte und die Server basierte Datenhaltung gegenübergestellt. Im Folgenden werden wesentliche Begriffe zum Bereich relationaler Datenbanken mit Hilfe von Beispielen und grafischen Abbildungen erläutert.

Abschließend wird an Hand der Anwendung TSN, die durch die Autoren über mehrere Jahre entwickelt wurde, gezeigt, welche Daten gespeichert werden sollten und wie ein daraus resultierendes Datenmodell aussehen muss, um auf die Daten schnell und effizient zugriffen zu können.

## Literatur

Sämtliche Definitionen und Begriffe sind ZEHNDER (1989) entnommen.

- SCHRÖDER, R., KLÖB, D., KROSCHEWSKI, K., STAUBACH, C. und MICKLICH, A. (2003):  
TSN – bundeseinheitliches Werkzeug zum Krisenmanagement von Tierseuchen  
In: MOLL, E. und STAUBER, T. (Hrsg.): Workshop Datenmanagement  
Berichte aus der Biologischen Bundesanstalt für Land- und Forstwirtschaft, Heft 119, S. 35-41
- ZEHNDER, C. A. (1989): Informationssysteme und Datenbanken  
B.G. Teubner Verlag, Stuttgart (ISBN 3-519-22480-1)



# **Datenmanagement am Beispiel des Informationssystems für die Zulassung von Pflanzenschutzmittel - INFOZUPF**

Worseck, S.

Biologische Bundesanstalt für Land- und Forstwirtschaft

(s.worseck@bba.de)

Das Informationssystem INFOZUPF wurde Anfang 1999 von einer Terminalanwendung auf eine Windows basierende Client-Server-Architektur auf der Basis von ORACLE umgestellt. Vor einer solchen Umstellung waren Richtungsentscheidungen bei der Wahl der äußeren und inneren Rahmenbedingen zu treffen, deren Wirkungen noch Jahre danach andauern. Ein Anliegen dieses Beitrages ist es, einige Grundsatzentscheidungen nochmals nachzuvollziehen und die positiven Erfahrungen den Fehlentscheidungen gegenüberzustellen.

An dieser Umstellung arbeiteten 10 DV-Mitarbeiter an zwei Standorten. Das Management der Programmentwicklung war eine wichtige Voraussetzung für das neue System. Nur mit klaren Richtlinien für die Programmierung, die Programmdokumentation, die Organisation des zyklischen Programmaustausches zwischen den Standorten Braunschweig und Kleinmachow, der Überführung von neuen Programmen in die Produktion, die Kennzeichnung von Programmversionen und die kontinuierliche Information der Nutzer über Programmänderungen konnte INFOZUPF mit ca. 270 Programmen und 110 Auswertungen aufgebaut werden.

Meilensteine waren Entscheidungen zum Umgang mit Kodern und deren Dekodierung, zum Einsatz des hierarchischen Kultur- und Schadorganismen-Kodes, zum System der Hilfsprogramme und zu Maßnahmen zur Wahrung der Datensicherheit und -integrität.

Für das laufende Informationssystem gewinnen die Aufgaben bezüglich des Managements der Datenbank und anderer Netzwerkressourcen immer mehr an Bedeutung.

## **1 Einleitung**

Das Bundesamt für Verbraucherschutz und Lebensmittelsicherheit (BVL) wurde mit Wirkung zum 1. November 2002 die zuständige nationale Behörde für die Zulassung von Pflanzenschutzmitteln in Deutschland. Zum Zeitpunkt des Vortrages nahm diese hoheitliche Aufgabe des Gesetzes zum Schutz der Kulturpflanzen – Pflanzenschutzgesetz (PflSchG) vom 1998 die Biologische Bundesanstalt (BBA) wahr. Aus diesem Grunde wird der damalige Zustand beschrieben.

### **1.1 Das Verfahren der Prüfung und Zulassung von PSM**

Aus der Vielzahl der hoheitlichen Aufgaben wird hier beispielhaft das damalige Prüfschema für die Zulassung von Pflanzenschutzmitteln dargestellt (Abb. 1).

Folgende Dinge sind dabei für die DV-technische Umsetzung relevant:

- Die schriftlich eingereichten Unterlagen des Antragstellers können mehrere Meter Akten umfassen. Die Firmen können diese Unterlagen zusätzlich auch auf CD (System CADDY) einreichen.
- Das Verfahren ist einer Terminkontrolle unterworfen.
- In dem Verfahren wird arbeitsteilig in 3 Behörden gleichzeitig die Vor- und Hauptprüfung durchgeführt.
- Die Prüfung und Bewertung erfolgt in der BBA gleichzeitig in unterschiedlichen Prüfebene und Prüfbereichen. Die Prüfbereiche können Nachforderungen bzw. Hinweise formulieren und die Zulassung an bestimmte Auflagen mit den entsprechenden Begründungen binden.
- In dem Verfahren werden Zwischenmitteilungen, Vorlagen für den Sachverständigen-Ausschuss (SVA) und der Bescheid generiert.
- Bei Ergänzungsanträgen bzw. bei Änderung der Sachlage werden Änderungsbescheide generiert.
- Die Biologische Bundesanstalt ist verpflichtet, ständig die aktuell gültigen Zulassungsdaten für die Öffentlichkeit bereitzustellen.

Die Durchführung des Zulassungsverfahrens ist ohne DV-Unterstützung nicht denkbar. Aus diesem Grund stehen Aspekte der Verfügbarkeit und des Frage/Antwort-Zeitverhaltens des Systems immer wieder im Vordergrund.

## 1.2 Gründe für die Umstellung der Terminalanwendung INFOZUPF

Das Zulassungsverfahren wurde bis Ende 1998 über zeichenorientierte Terminalanwendungen mit dem Datenmanagementsystem RdB der Firma DEC unter dem Betriebssystem OpenVMS unterstützt. Diese Anwendungen basierten auf den Programmiersprachen Rally, Pascal, Cobol, C, SQL und DEC-Forms.

Folgende Punkte sprachen 1995 für eine Umstellung des damaligen Systems:

- RdB lief nur unter den Betriebssystemen OpenVMS und DEC-UNIX und damit nur auf Hardwareplattformen der Firma Digital/Compaq. Durch einen Umstieg auf ein alternatives Datenbankmanagementsystem sollte die enge Herstellerabhängigkeit durchbrochen werden.
- Die Programmierung in den oben genannten Sprachen war sehr arbeitsintensiv. Aus diesem Grund war die Anzahl der Programme gering und es gab eine große Liste unerfüllter Nutzer-Wünsche.
- Die Einvernehmensbehörden schickten ihre Bewertungen in Word. In der BBA wurden diese Wordtexte dann durch die Mitarbeiter mit den beschränkten Möglichkeiten des benutzten ASCII-Editors der Terminalanwendung mit hohem Zeitaufwand in die Datenbank Zeile für Zeile überführt. Für die Erstellung der Prüfberichte und Wirkstoff-Monografien wurde eine moderne Textverarbeitung benötigt.
- Durch Umstellung auf eine Client/Server-Technologie sollten die Server entlastet werden und primär nur noch für die Datenbankaufgaben eingesetzt werden.
- Den Anwendern standen bei kodierten Eingaben keine Online Hilfen in Form von Auswahllisten zur Verfügung.
- Die Umstellung auf den hierarchischem BAYER-Kultur- und Schadorganismenkode war längst überfällig und wurde aus Kapazitätsgründen immer wieder hinausgeschoben.

## 2 Wesentliche Entscheidungen vor der Umstellung von INFOZUPF

### 2.1 Grundsatzentscheidungen

Eine Umstellung musste so konzipiert werden, dass mit dem vorhandenen Personal das Laufen der alten Anwendungen gesichert und gleichzeitig die neuen Anwendungen erstellt und getestet wurden. In der Zentralen EDV-Gruppe hatten unter der Leitung von Dr. Schwartz daran Anteil: Herr Jeske, Herr Dr. Warseck, Herr Sellmann, Frau Schmidt, Herr Dang, Herr Gils, Herr Liersch, Frau Gewinnus, Frau Haecker und Frau Preuß. Daneben gab es viele Mitarbeiter in den Fachgruppen, die diese Umstellung fachlich begleiteten.

#### 2.1.1 Oracle als Datenbanksystem

1995 wurde das Datenmanagementsystem RdB von der Firma Digital/Compaq an die Firma ORACLE verkauft. Das von ORACLE angebotene Upgrade-Programm für den Umstieg von RdB auf ORACLE war der Anlass, in der BBA nochmals über die DV-technische Unterstützung des Zulassungsverfahrens nachzudenken und sich gegebenenfalls neu zu positionieren. Auf Grund dieser Analyse wurde 1995 die Grundsatzentscheidung getroffen, INFOZUPF auf eine Client/Server-Architektur mit dem Datenbanksystem ORACLE umzustellen.



### 2.1.2 MS-Windows basierende Clients

Diese Entscheidung war marktorientiert. Dass Windows kein optimales Betriebssystem ist, wusste man schon zum Zeitpunkt der Entscheidung (1995). Die Möglichkeit des Einsatzes von alternativen Systemen war und ist beschränkt (OpenSource). Allerdings ist der Kostenfaktor „Client“ heute erheblich und bedarf einer kompetenten Einkaufsstrategie.

Die in Abb. 2 dargestellte INFOZUPF-Architektur umfasst folgende Merkmale.

- Auf dem PC sind das Textverarbeitungssystem Word und die Runtime von ORACLE zu installieren und zu pflegen.
- Die Daten liegen auf einem zentralen Server in Braunschweig. Alle Änderungen erfolgen nur in dieser Datenbank. Die entfernten Standorte arbeiten über Standleitungen.
- Die Anwendungsprogramme, Auswertungen, Hilfen und Bibliotheken liegen auf einem Server je Standort.
- Eine Spezialkonstruktion wird derzeit durch das Bundesamt für gesundheitlichen Verbraucherschutz und Veterinärmedizin (BgVV) als Einvernehmensbehörde getestet. Dabei greifen die Nutzer des BgVV über eine Terminal-Sitzung auf einen entfernten Rechner zu.

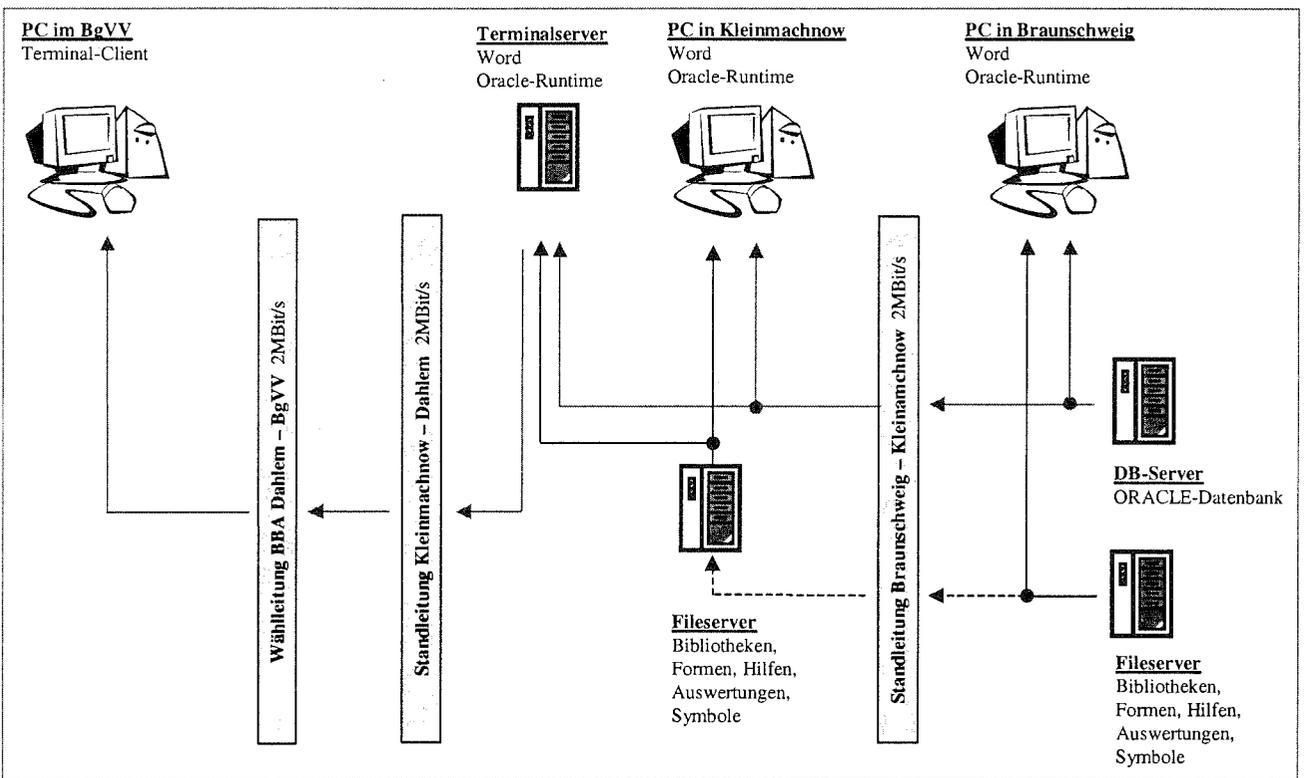


Abb. 2: INFOZUPF-Architektur

### 2.1.3 OLE-Verarbeitung

Ein Hauptbestandteil von INFOZUPF sollte die zeitgleiche Erstellung und Pflege von Textbausteinen in Word sein, die dann zu den Hauptprodukten des Verfahrens zusammengeführt werden (Monografie, Prüfbericht, Zwischen-Mitteilung, Bescheid, Änderungsbescheid). Wir hatten uns 1995 für die Speicherung der Wordtexte in LONGRAW-Felder unter ORACLE entschieden. Die Anwendungen mussten dann über OLE-Verarbeitung auf die Wordtexte zugreifen.

Aus heutiger Sicht war diese Variante eine Fehlentscheidung, da

- mit der Festlegung auf LONGRAW-Feldern keine Replikation möglich war und damit nur eine Datenbank in Braunschweig stehen konnte und
- die Entwicklungstools für Intranet-Anwendungen damals noch keine OLE-Verarbeitung konnten. Deshalb musste eine klassische Client/Server-Architektur eingesetzt werden.
- Heute wissen wir, dass etliche nicht mehr lesbare OLE-Objekte in der Datenbank gespeichert wurden, die zu großem Unmut bei den Nutzern führten.  
Erschwerend kommt noch hinzu, dass ORACLE in der Datenbank um das Wordobjekt noch weitere Informationen speichert, die nicht offengelegt sind, so dass man nicht in der Lage ist, den Rest-Inhalt des „defekten“ Objektes auszulesen.

In einer Grundsatzentscheidung haben wir festgelegt, die LONGRAW-Felder auf die ab ORACLE 8 verfügbaren CLOBS umzustellen. In diesen Feldern wird dann der Wordtext im RTF-Format (MICROSOFT 2001) gespeichert.

## 2.1.4 ORACLE-Developer2000 als Entwicklungstool

### ORACLE-Developer2000 Formen

Grundsätzlich war die Entscheidung zur Programmierung mit Forms für Client/Server-Anwendungen richtig.

- Die Anwendungen haben ein schnelles Frage-Antwort-Zeit-Verhalten, wenn richtig programmiert wird.
- Das Frage-Antwort-Zeit-Verhalten verschlechtert sich über eine Standleitung von 128 kBit bei intensiver Nutzung von Master-Detail-Beziehungen, berechneten Feldern und Post-Query-Triggern.
- Mit Form 6.0 standen lang ersehnte Funktionalitäten zur Verfügung (Darstellung von Registerkarten, Hierarchien und das Setzen von Display-Eigenschaften einzelner Zellen). Doch der Versionsumstieg von Forms 4.5 auf Forms 6.0 war im Detail schlimmer als gedacht. Wir fanden diverse Bugs und mussten hoffen, dass dafür dann entsprechende Patches zur Verfügung gestellt wurden.

Dennoch darf man nicht verschweigen, dass es auch nicht analysierbare Abstürze von Oracle-Formen durch die sehr enge Verzahnung der ORACLE- und Microsoft-Komponenten gab. Der ORACLE-Support konnte nicht helfen und tat dies mit dem pauschalen Hinweis auf nicht richtig implementierte Betriebssystemkomponenten unter Windows 9x ab. Weiterhin weigerte sich ORACLE, die in diesem Zusammenhang stehenden relevanten Schnittstelle offen zu legen.

### ORACLE-Developer2000 Auswertungen und alternative Auswertungstools

Nach kurzen Tests war klar, dass der ORACLE-Reports Builder OLE-Objekte, die selbst länger als eine Seite sind, nicht ausgeben kann. Insofern mussten alternative Auswertungstools genutzt werden.

- Das Entwicklungswerkzeug **Reports Builder 6i** ist ein mächtiges Tool, das keine Wünsche in der Datenaufbereitung für einen Druck offen lässt. Der Builder läuft allerdings instabil, die gebildeten Auswertungen sind dagegen stabil.  
In INFOZUPF gibt es ca. 120 Auswertungen, die mit dynamisch zusammenstellbaren Filterbedingungen gestartet werden. Will man aber das Ergebnis einer Auswertung nicht drucken, sondern z.B. als RTF-Datei konvertiert abspeichern und in Word nacharbeiten, dann ist das eine Sackgasse, da es sich immer um Einzelobjekte handelt (Abb. 3)
- Wenn als Ergebnis der Auswertung ein Wordtext benötigt wird, werden z.Z. drei alternative Wege genutzt:
  - a) Ausfüllen von Textfeldern in Word-Vorlagen über **OLE-Verarbeitung** (Word-Basic). D.h. Word verarbeitet die von der ORACLE-Anwendung ausgesendeten Word-Basic-Befehle, wie z.B. „Datei öffnen“, „Datei speichern unter“, „Gehe zu Textmarke ...“, „Füge Text ... ein“. Diese Programmierung ist einfach, sehr anfällig (Speicherbedarf, ...), langsam, nur für eine bestimmte Wordversion mit WordBasic gültig und die Formatierungsmöglichkeiten sind sehr beschränkt. Komplexe Zusammenhänge lassen sich schlecht in einer einzigen Vorlage bedenken (Abb. 4)

- b) Ansteuerung von Word über **Visual-Basic**. Die Funktionalität ist ähnlich. Allerdings ist der Programmieraufwand höher (Abb. 5)
- c) Die Ausgabe als **RTF-Text** erscheint als eine Alternative, weil man über diesen Weg alle Formatierungsmöglichkeiten des RTF-Syntax verwenden kann. Die Programmierung ist ca. 5x komplizierter und wahrscheinlich noch wartungsunfreundlicher!  
Dennoch bietet sich die Ausgabe als RTF-Text bei sehr komplexen Zusammenhängen an, da sehr anspruchsvolle Texte erstellt werden können (Abb. 6, Abb. 7)
- Die Ausgabe von **Excel-Datenblättern** über OLE-Verarbeitung klappt ausgesprochen gut.
- Einzelne Standardauswertungen laufen nächtlich auf dem Server über SQL ab, deren Ergebnisse in **HTML-Dateien** gespeichert werden.
- Es bestand noch kein Bedürfnis, Auswertungen über **Graphics** zu programmieren.
- Für weitergehende Auswertungen steht der Zugriff über **ODBC** zur Verfügung.

**BIOLOGISCHE BUNDESANSTALT FÜR LAND - UND FORSTWIRTSCHAFT  
BRAUNSCHWEIG**  
ABTEILUNG FÜR PFLANZENSCHUTZMITTEL UND ANWENDUNGSTECHNIK  
**INFORMATIONSSYSTEM INFOZUPF** DATUM : 2002-04-26

**Übersicht zu Anwendungen**

Mit "\*" gekennzeichnete AWG sind nicht freigegeben!

	Zul.-status	Eins.-gebiet	Wirk.-bereich	Anw.-bereich																																													
<b>ANWENDUNGS GEBIET</b>																																																	
<b>Einschränkung :</b>	Antragsteller = BAY																																																
	Anwendungsbereich like F%																																																
	Anwendungstechnik Liste SXX,SST																																																
	AWG-Freigabe (J / N) = J																																																
<b>Sortierung :</b>	Kennnummer																																																
	Anwendungsgebietsnummer																																																
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">000023-67</td> <td style="width: 40%;">Rasen-Hedomat</td> <td style="width: 10%;">Zul.-stat : IIII</td> <td style="width: 15%;">Zul.-erstm. : 1987-05-20</td> <td style="width: 15%;">Zul.-ende : 1989-12-31</td> </tr> <tr> <td>Antragart : ZA</td> <td>Antragsteller : BAY</td> <td colspan="3">Wirkbereich Mittel : H</td> </tr> <tr> <td>Wirkstoff : 0074</td> <td>MCPA</td> <td colspan="3"></td> </tr> <tr> <td></td> <td>0218 Dicamba</td> <td colspan="3"></td> </tr> <tr> <td>00-001</td> <td>gegen: Zweikeimblättrige Unkräuter in: Sommerweizen, Sommerroggen, Sommergerste, Hafer</td> <td>NN</td> <td>A</td> <td>H FX</td> </tr> <tr> <td>00-004</td> <td>gegen: Zweikeimblättrige Unkräuter in: Wiesen, Weiden</td> <td>NN</td> <td>R</td> <td>H FX</td> </tr> <tr> <td>00-005</td> <td>gegen: Zweikeimblättrige Unkräuter in: Rasen</td> <td>NN</td> <td>A</td> <td>H FX</td> </tr> <tr> <td>02-001</td> <td>gegen: Zweikeimblättrige Unkräuter in: Landwirtschaftlich-nutzungsflächen, Grasflächen</td> <td>NN</td> <td>A</td> <td>H FX</td> </tr> <tr> <td>00230-00</td> <td>"*" erneut</td> <td></td> <td></td> <td></td> </tr> </table>					000023-67	Rasen-Hedomat	Zul.-stat : IIII	Zul.-erstm. : 1987-05-20	Zul.-ende : 1989-12-31	Antragart : ZA	Antragsteller : BAY	Wirkbereich Mittel : H			Wirkstoff : 0074	MCPA					0218 Dicamba				00-001	gegen: Zweikeimblättrige Unkräuter in: Sommerweizen, Sommerroggen, Sommergerste, Hafer	NN	A	H FX	00-004	gegen: Zweikeimblättrige Unkräuter in: Wiesen, Weiden	NN	R	H FX	00-005	gegen: Zweikeimblättrige Unkräuter in: Rasen	NN	A	H FX	02-001	gegen: Zweikeimblättrige Unkräuter in: Landwirtschaftlich-nutzungsflächen, Grasflächen	NN	A	H FX	00230-00	"*" erneut			
000023-67	Rasen-Hedomat	Zul.-stat : IIII	Zul.-erstm. : 1987-05-20	Zul.-ende : 1989-12-31																																													
Antragart : ZA	Antragsteller : BAY	Wirkbereich Mittel : H																																															
Wirkstoff : 0074	MCPA																																																
	0218 Dicamba																																																
00-001	gegen: Zweikeimblättrige Unkräuter in: Sommerweizen, Sommerroggen, Sommergerste, Hafer	NN	A	H FX																																													
00-004	gegen: Zweikeimblättrige Unkräuter in: Wiesen, Weiden	NN	R	H FX																																													
00-005	gegen: Zweikeimblättrige Unkräuter in: Rasen	NN	A	H FX																																													
02-001	gegen: Zweikeimblättrige Unkräuter in: Landwirtschaftlich-nutzungsflächen, Grasflächen	NN	A	H FX																																													
00230-00	"*" erneut																																																

**Abb. 3:** Beispielauswertung mit ORACLE-Reports in der Druckansicht und als RTF-Text gespeichert. Jedes Feld ist ein Objekt - ein nachträgliches Editieren wird unmöglich.

```

APPLICATION      OLE2.OBJ_TYPE;
PROCEDURE DOKUMENTOEFFNEN (APPLICATION OLE2.OBJ_TYPE, ARGUMENT VARCHAR2) IS
  ARGS OLE2.LIST_TYPE;
BEGIN
  ARGS := OLE2.CREATE_ARGLIST;
  OLE2.ADD_ARG(ARGS, ARGUMENT);
  OLE2.INVOKE(APPLICATION, 'FILEOPEN', ARGS);
  OLE2.DESTROY_ARGLIST(ARGS);
END;

BEGIN
  APPLICATION := OLE2.CREATE_OBJ('WORD.BASIC');
  DOKUMENTOEFFNEN (APPLICATION, 'D:\VORLAGEN\MUSTER.DOC');
  ....
  SCHLIESSEWORD (APPLICATION);
END;
    
```

**Abb. 4:** Beispielprozedur zum Öffnen einer Datei mit Wordbasic-Befehlen

```

APPLICATION OLE2.OBJ_TYPE;
PROCEDURE DOKUMENTOEFFNEN (APPLICATION OLE2.OBJ_TYPE, ARGUMENT VARCHAR2) IS
  ARGS OLE2.LIST_TYPE;
  MYDOCUMENTS OLE2.OBJ_TYPE;
  MYACTIVEDOCUMENT OLE2.OBJ_TYPE;
BEGIN
  MYDOCUMENTS:=OLE2.GET_OBJ_PROPERTY(APPLICATION, 'DOCUMENTS');
  ARGS:= OLE2.CREATE_ARGLIST;
  OLE2.ADD_ARG(ARGS, ARGUMENT);
  OLE2.INVOKE(MYDOCUMENTS, 'OPEN', ARGS);
  MYACTIVEDOCUMENT:=OLE2.GET_OBJ_PROPERTY(APPLICATION, 'ACTIVEDOCUMENT');
  OLE2.RELEASE_OBJ(MYACTIVEDOCUMENT);
  OLE2.RELEASE_OBJ(MYDOCUMENTS);
  OLE2.DESTROY_ARGLIST(ARGS);
END;

BEGIN
  APPLICATION:=OLE2.CREATE_OBJ('WORD.APPLICATION');
  DOKUMENTOEFFNEN(APPLICATION, 'D:\VORLAGEN\MUSTER.DOC');
  ...
  SCHLIESSEWORD(APPLICATION);
END;
    
```

Abb. 5: Beispielprozedur zum Öffnen einer Datei mit Visualbasic-Befehlen

Rechtsbehelfsbelehrung  
 Gegen diesen Bescheid kann innerhalb eines Monats nach Bekanntgabe Widerspruch erhoben werden. Der Widerspruch ist bei der Biologischen Bundesanstalt, Messeweg 11/12, 38104 Braunschweig, schriftlich oder zur Niederschrift einzulegen.

In Vertretung

Dr. H.-G. Nolting

```

RTF_ZEILE(
'\pard\keepn\widctlpar {Rechtsbehelfsbelehrung\par 'Gegen diesen Bescheid kann innerhalb eines Monats nach
Bekanntgabe Widerspruch erhoben werden.Der Widerspruch ist bei der Biologischen Bundesanstalt, Messeweg 11/12,
38104 Braunschweig, schriftlich oder zur Niederschrift einzulegen. \par \par In Vertretung \par \par \par
||:UNTERSCHRIFT||'\pard\par }');
    
```

Abb. 6: Um obigen Textausschnitt in Word zu erhalten, muss eine entsprechende RTF-Zeile geschrieben werden.

		g/kg		Anwachsen; bei Infektionsgefahr bzw. ab Warndienstinweis					
	WP	500 g/kg	spritzen	Nach dem Anwachsen; bei Infektionsgefahr bzw. ab Warndienstinweis	3 ; -	0,083	600	0,5	14
	WP	500 g/kg	spritzen mit Dreidüsendügel	Beginn der Blüte und Mitte der Blüte und Ende der Blüte	3 ; -	0,05	2000	1	10
	WP	500 g/kg	spritzen oder sprühen	Bei Infektionsgefahr bzw. ab Warndienstinweis	3 ; -				28
				Basisaufwand:		0,009 - 0,038	400 - 1600	0,15	
				ES 61:		0,019 - 0,075	400 - 1600	0,3	
				ES 71:		0,028 - 0,113	400 - 1600	0,45	
				ES 75:		0,038 - 0,15	400 - 1600	0,6	
	WP	500 g/kg	spritzen	Bei Befallsgefahr	- ; - (7 bis 14 Tage)			[Mittel:0,1 %]	XN
	WP	500 g/kg	spritzen	Bei Befallsgefahr	- ; - (7 bis 14 Tage)			[Mittel:0,1 %]	XN

Abb. 7: Ausschnitt einer Anwendungsliste  
 Diese Tabelle wurde mit RTF-Syntax mit folgenden Besonderheiten geschrieben: Verbundene Zellen, Seitenumbruch nur nach vollständigen Anwendungen und Nutzung von verborgenem Text.

## 2.2 Management – Datenbankdesign / Metadaten

Für das Datenbankdesign wurde der ORACLE-Designer eingesetzt. Aus der großen Funktionsvielfalt des Designers wurden folgende Arbeitsschritte genutzt:

- Erstellung des Datenbankentwurfs,
- Ausgabe von E/R-Diagrammen zur Diskussion des Datenbankentwurfs und
- Erstellung von Scripten für das Anlegen der Tabellen und Triggern.
- Vor der Programmierung wurde mit dem Moduldesign eine Abgrenzung der Aufgaben mit den wesentlichsten Übergabeparametern vorgenommen.

Nach der Datenbank-Umstellung wird der Designer nur noch zur Pflege der E/R-Diagramme als Datenbankdokumentation und für die Integration weiterer Projekte eingesetzt. In der Phase der Programmierung wurde von den Programmierern beklagt, dass sie kein Tool haben, um sich die Datenbankstruktur für ein Teilprojekt vollständig anzusehen (Tabellendefinition, DB-Trigger, Constraints, ...).

## 2.3 Management – Programmentwicklung

Das klassische Phasenmodell der Softwareentwicklung kann man optimal in Anwendungsbereichen mit

- in sich abgeschlossenen Aufgaben und bei
- möglichst zeitlich stabilen Rahmenbedingungen

einsetzen.

In der Realität haben große Projekte aber eine Eigendynamik, so dass nicht von zeitlich stabilen Rahmenbedingungen ausgegangen werden kann. Das Phasenmodell ist somit nur bei „Momentaufnahmen“ vom Projekt anwendbar.

Unsere Produkte sind die Anwendungen und die sind so gut, wie es gelungen ist,

- das Gesamtprojekt in sinnvolle Teilprojekte aufzuteilen,
- die Teilprojekte parallel durch Mitarbeiter programmieren zu lassen,
- die Kreativität der "Softwareentwickler" nicht nach außen sichtbar werden zu lassen,
- das Frage-Antwort-Zeit-Verhalten zu optimieren,
- Programmänderungen zu testen,
- eine kurze Reaktionszeit bis zur Korrektur von Programmfehlern im laufenden Betrieb einzuhalten und
- wie die Stabilität der Anwendung im "realen Leben" ist.

Die IT-Anwendung INFOZUPF hat einen solchen Umfang, dass diese nur mit einer Gruppe von Entwicklern erstellt und gepflegt werden kann. Deshalb mussten vor Beginn der Programmentwicklung zahlreiche Festlegungen getroffen werden.

### 2.3.1 Programmiervorgaben

Die wesentlichsten Programmiervorgaben waren:

- **Layout-Vorgaben** (Schriftarten, Farben, Schaltflächen, ...)
- Vorgaben zu **Nutzeraktionen**,
- zu **Programmbibliotheken** und Symbolen (PLL, OLB, MMB, ICO),
- zu **Programmbezeichnungen**,

Die Systematik in der Bezeichnung der Programme hat sich prinzipiell bewährt, obwohl sich in der Zwischenzeit Überschneidungen der zunächst klaren Untergliederung eingestellt haben.

S\_ Spezielle Suchprogramme, die selbständig laufen können oder als Unterprogramm Rückgabewerte an das übergeordnete Programm zurückgeben können.

I\_ Informationsprogramme, die nur lesend komplette Inhalte von Tabellen darstellen.

B\_ Programme zum Bearbeiten eines oder mehrerer Datensätze einzelner Tabellen.

R\_ Reports über ORACLE-Reports

... Insbesondere komplexe Auswertungen, wie z.B. GENERIERE\_BESCHEID

- zum zyklischen **Programm-Austausch**,  
Organisation des zyklischen Programm-Austausches (Quellen und ausführbare Programme) zwischen den Standorten Braunschweig und Kleinmachnow und Überführung von neuen Programmen in die Produktion.
- und zu **Programmversionen** - Archiv der Quellen.  
Um alte Quellen nicht zu überschreiben, wird an den Programmnamen der Quelle mit Unterstrich getrennt die jeweilige Versionsnummer angehängt. Die Versionsnummer erscheint ebenfalls bei der Abarbeitung in der Titelzeile des Runtime-Fensters. Muss eine Programm-Änderung wieder rückgängig gemacht werden, so kompiliert man die nächst ältere Version nochmals und stellt diese in die Produktion ein.  
Damit die Nutzer über die Änderungen informiert werden, werden die Änderung in einer DB-Tabelle dokumentiert. Beim Start eines Programms wird der Nutzer informiert, falls Programmänderungen vorliegen. Dann erscheint in der Symbolleiste ein rotes Dreieck mit Ausrufezeichen.

### 2.3.2 Modulares Konzept

Universell einsetzbare Funktionen und Prozeduren ohne Datenbankzugriffe wurden in PLL-Bibliotheken abgelegt. Alle universell einsetzbaren Funktionen und Prozeduren mit Datenbankzugriffen befinden sich jedoch direkt in der Datenbank. Diese DB-Funktionen und DB-Prozeduren sind so getestet, dass optimale Zugriffsstrategien durch ORACLE genutzt werden. Damit wird ein schneller Datenzugriff gewährleistet.

Die nächst höhere Ebene wiederverwertbarer Module sind die Forms-Module selbst. Z.B. sind die Hilfsprogramme von INFOZUPF so konzipiert, dass sie selbständig laufende Formen sind, die aber gleichzeitig auch als Unterprogramme genutzt werden können. Werden sie als Unterprogramm genutzt, werden globale Variablen als Suchparameter übergeben und ein oder mehrere Rückgabewerte beim Verlassen des Unterprogramms gesetzt. Das wohl am häufigsten genutzte Hilfsprogramm ist das Programm „Suche von Kodern“. Steht man in einem kodieren Eingabefeld, kann man mit Doppelklick in dem Feld dieses Unterprogramm für die spezifische Kodelisten-Nummer aktivieren. Hat man den entsprechenden Code im Hilfsprogramm gefunden, verlässt man es durch Doppelklick auf diesen Code und überführt ihn damit in das Eingabefeld.

### 2.4 Management – Dokumentation

Die Programmierer schreiben die Dokumentationen in Word (Druck-Version), die dann als HTML-Text konvertiert für die Online-Dokumentation abgelegt wird.

Jedes Forms-Programm wird einzeln dokumentiert. Drückt der Nutzer in der Applikation die Hilfetaste F1, wird der Formsname der aktuellen Applikation ausgelesen und der entsprechende HTML-Text angezeigt. Geplant war eigentlich der Druck eines kompletten Manuals über die Word-Mechanismen mit „Filial-Dokumenten“ und automatischer Indizierung. Word ist allerdings mit einem Umfang von 280 Einzeldokumenten und ca. 200 MB eindeutig überfordert.

### 2.5 Management – Kode

In INFOZUPF werden viele Eingaben kodiert vorgenommen. Die Vorteile sind:

- einfache, verkürzte Eingabe,
- Möglichkeit der Validierung der Eingaben,
- bessere und schnellere Suche nach Daten.

### 2.5.1 Kode und deren Dekodierung

Aus diesem Grund wurden in einer Tabelle KODELISTE alle Kodelisten mit einer Kodelisten-Nummer definiert. In der Tabelle KODE liegen zu jeder Kodelisten-Nummer alle Kode mit deren Dekodierungen. Es besteht ebenfalls die Möglichkeit, Dekodierungen in unterschiedlichen Sprachen zu hinterlegen. Über die Tabelle KODELISTE\_FELDDNAME wird festgelegt, welche Kodelisten-Nummer für die Eingabe in einem Feld einer Tabelle genutzt werden muss.

Die meisten Anfragen für diese Tabellen werden über zwei zentrale DB-Funktionen realisiert:

- Mit der Funktion LISTE sucht man sich die Kodelisten-Nummer für ein Feld einer Tabelle und
- mit der Funktion TEXT bekommt man für einen Kode die Dekodierung in einer Sprache aus einer bestimmten Kodeliste.

### 2.5.2 Kultur- und Schadorganismen-Kode

In INFOZUPF hat die Arbeit mit Kultur- und Schadorganismen-Koden einen hohen Stellenwert. Bis zur Umstellung auf ORACLE wurde ein BBA-spezifischer Kultur- und Schadorganismen-Kode genutzt. Die Schwächen dieser BBA-Kode hatte man schon vor ca. 10 Jahre erkannt, hatte aber keine Kraft, das gewachsene System umzustellen. Die Kodierungen waren nach den Bedürfnissen für die Ausgabe im Pflanzenschutzmittelverzeichnis entstanden und hatten folgende Schwachpunkte:

- der Kode war nicht hierarchisch und
- die ausgenommenen Kulturen wurden hart kodiert.

Anhand der Abb. 8 ist es nachvollziehbar, dass z.B. eine Abfrage: „Suche mir alle Anwendungen in Gemüse Kohl“ mit dieser Datenbasis nicht zu beantworten ist. Dies ist ein typisches Beispiel dafür, dass man sich beim Entwurf einer Datenbank an den potentiellen Abfragen orientieren muss.

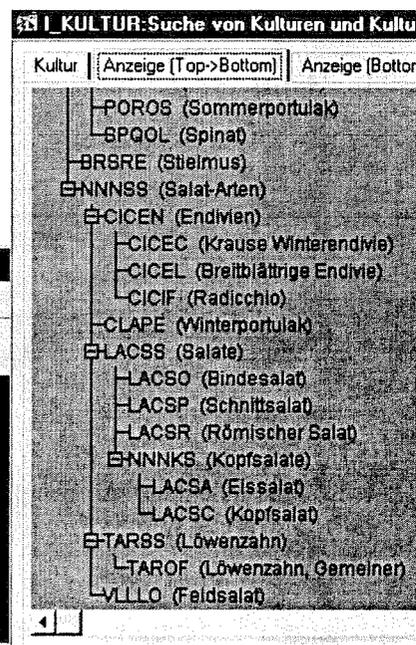
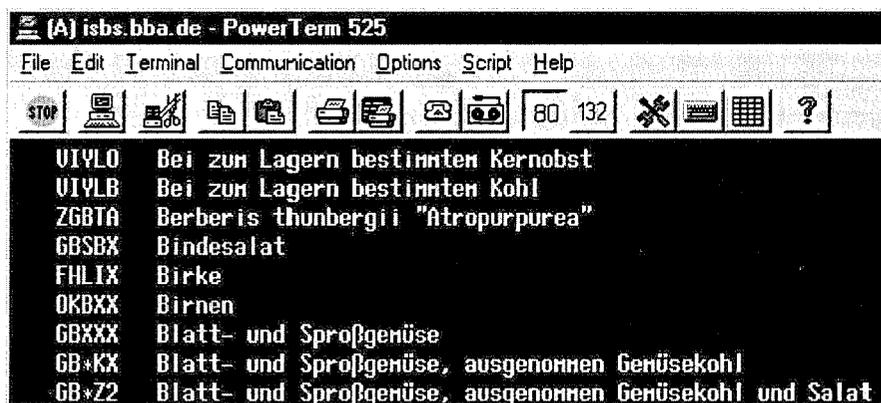


Abb. 8: Ausschnitte aus der alten BBA-Kultur-Kodeliste und dem neuen hierarchischen Kulturcode

## 2.6 Management – Datensicherheit

U.a. wurden folgende ORACLE- spezifischen Maßnahmen ausgenutzt:

### Rollenkonzept

- Passwortschutz der Nutzer,
- Rolle = Aufgabenprofil (jeder Rolle werden die notwendigen Datenbankrechte übertragen),
- Zuordnung der Nutzer zu den Rollen.

## Transaktionen

- Die Transaktionssicherheit wird vollständig ORACLE überlassen.
- Häufig wurde von uns in den Formen zusätzlich vor der Datenänderung dieser Datensatz mit der Option "for update" gelesen. Kann der Datensatz nicht reserviert werden, kommt eine Fehlermeldung.
- Die Trennung in nur lesende und schreibende Programmen brachte den Vorteil der klaren Abgrenzung von Transaktionen. Der Bedien-Komfort des Programms ließ aber durch die vielen Fenster und Aktionen zu wünschen übrig. Neuere Programme weichen von dieser harten Vorgabe ab.

## Datenverlust

- Speicherung auf gespiegelten Platten,
- nächtliche vollständige Sicherung der Daten,
- am Tage Protokollierung aller Änderungen in REDO-LOGs

## Nachvollziehbarkeit

- Datum und Nutzer werden in fast allen Tabellen über Datenbank-Trigger automatisch beim schreibenden Zugriff geändert (Abb. 9).
- In wenigen Anwendungen wird eine Historie geführt.

```
CREATE OR REPLACE TRIGGER TRI_AUFLAGE_U
BEFORE UPDATE
ON AUFLAGE
FOR EACH ROW
BEGIN
:NEW.NUTZER:=NUTZERNAME;
:NEW.DATUM:=SYSDATE;
:NEW.AUFLAGE_BEREICH:=SUBSTR(:NEW.AUFLAGE,1,2);
END;
```

**Abb. 9:** Beispielscripte für einen Datenbanktrigger

Zusätzlich zu den ORACLE-spezifischen Möglichkeiten wurden BBA-interne Schutzmechanismen entwickelt und eingesetzt. Damit wurde z.B. ein Schutz von Datensätzen zwischen den unterschiedlichen Prüfgebieten realisiert.

## 2.7 Management – Datenintegrität

Die Datenintegrität ist ein Schlüsselparame-ter für die Qualität der Datenbank.

- Constraints validieren vor dem entgeltigen Speichern die Daten.
- Automatisches Löschen von abhängigen Datensätzen bei 1:N Verknüpfungen.
- Vermeiden Sie unbedingt die Speicherung von redundanten Informationen. Der Aufwand zum Datenabgleich ist im Fehlerfall sehr hoch.
- Besser ist die Nutzung von Sichten (Views).

Beispielscripte sind in der Abb. 10 vorgestellt.

Sollte es für die Optimierung der Anwendungen notwendig sein, doch redundante Informationen (vielleicht anders aufbereitet) abzuspeichern, sollte dies in den Datenbanktriggern geschehen (Abb. 9).

```

ALTER TABLE AWG_SUCHE
  ADD (CONSTRAINT AWG_SUCHE_PK PRIMARY KEY (RECHERCHE_NR)
  USING INDEX
  STORAGE ( INITIAL 1000000 NEXT 200000 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 100000 )
  TABLESPACE TABSP_AWG);

ALTER TABLE AWG_SUCHE_SCHADORG ADD CONSTRAINT
  AWG_SUCHE_SCHADORG_FK FOREIGN KEY
  (RECHERCHE_NR) REFERENCES AWG_SUCHE
  (RECHERCHE_NR) ON DELETE CASCADE;

ALTER TABLE RU_VERS
  ADD (CONSTRAINT AVCON_183072_NACHB_000 CHECK (NACHBAU IN ('J', 'N')));

CREATE OR REPLACE VIEW V_AWG_ZULASSUNG AS SELECT
  A.AWG_ID
  BESCHEID_VOM
  ZUL_ENDE
  ZUL_STATUS
  VORGES_ZUL_ENDE
  BESCHEID_ENTWURF
  SUBSTR (NVL (ANTRAGART, 'Z?'), 1, 1) BESCHEID_TYP,
  ANTRAGSTELLER
  FROM AWG A, ANTRAG B, AWG_ZULASSUNG C
  WHERE
    A.KENNR =B.KENNR AND
    A.ANTRAGNR=B.ANTRAGNR AND
    A.AWG_ID =C.AWG_ID;

```

Abb. 10: Beispielscripte in INFOZUPF für Constraints und eine View

### 3 Datenübernahme RdB → ORACLE

#### 3.1 Auf dem Server unter VMS

Es wurden für einzelne Teilprojekte unterschiedliche Verfahren zur Datenübernahme ausgenutzt. Die meisten Tabellen wurden über folgende Schritte geladen (Abb. 11):

- Export auf ASCII-File,
- Import über ORACLE SQL-Loader und
- gegebenenfalls Update in SQL.

Feststehende Daten können auch direkt über das Controlfile (Abb. 12) geladen werden.

```

$ USER      ="DB_ADMI"
$ PASSWORD  ="XXXXXXX"
$ SERVICE   ="TZBS"
....
$ DTR @AUFLAGE_AUFLG
$ SQLLOAD USERID='USER'/'PASSWORD'@'SERVICE, CONTROL=AUFLAGE_AUFLG.CTL
$ SQLPLUS 'USER'/'PASSWORD'@'SERVICE'@UPDATE_AUFLAGE.SQL
....

```

Abb. 11: VMS – Kommando-Prozedur zum Entladen, Laden und Ändern von Daten

```

OPTIONS(ERRORS=50000)
LOAD DATA INFILE *
REPLACE INTO TABLE LUECKE_MITTEL
TRAILING NULLCOLS (
  DATUM          SYSDATE,
  NUTZER         CONSTANT '?',
  LUECKE_SEQ     TERMINATED BY X'9',
  BEARBEITET_AM TERMINATED BY X'9' "TO_DATE (:BEARBEITET_AM, 'DD.MM.YYYY HH24:MI:SS')",
  KENNR         TERMINATED BY X'9',
  STATUS        TERMINATED BY WHITESPACE )
BEGINDATA
1          003965-00      1
1          023701-00      1
3          004305-00      3
3          004560-00      2

```

Abb. 12: Controlfile des SQL-Loaders zum Laden von Daten.

In diesem Beispiel beginnen die Datensätze nach dem Schlüsselwort BEGINDATA

Der Vorteile dieses Verfahrens sind:

- Dokumentation aller zu übernehmenden Daten auf einem ASCII-File. Dies ist hilfreich für Nachfragen.
- Während des Exportes können schon einzelne Konvertierungen vorgenommen werden (z.B. fiktive NULL-Werte löschen, Datumskonvertierungen, ...), um sich der neuen DB-Struktur anzunähern.
- Der SQL-Loader war ein wesentliches Tool, um das Laden durchzuführen und die Vollständigkeit und die Richtigkeit des Ladens zu protokollieren. Werden DB-interne Constraints verletzt, wird nachvollziehbar der entsprechende Datensatz vom Import ausgeschlossen und im sogenannten BAD-File gespeichert.

Die meiste Zeit musste investiert werden, um den Ausnahmen beim Laden nachzugehen. Dabei kamen vielfach Unzulänglichkeiten in der Datenbasis zum Vorschein. Z.B. gab es Datensätze, die die Nutzer nicht mehr sahen. Bevor solche Sätze jedoch gelöscht werden konnten, mussten die Fachgruppen jede Ausnahme prüfen.

### 3.2 Auf dem PC unter Windows mit SQL

Einige Teilprojekte wurden erst nachträglich nach ORACLE überführt. Dazu wurde alternativ folgender Weg genutzt:

- Kopieren der zu exportierenden Tabelle über ODBC von RdB in eine ORACLE – Hilfstabelle und
- Konvertierung der Daten in die Zieltabelle.

Dieser Weg erscheint einfacher, hat aber auch seine Grenzen:

- Alle Daten müssen vom Server über die ODBC-Schnittstelle des PCs geholt werden und wieder auf den Server zurückgeschrieben werden. Dies ist ein echtes Nadelöhr.
- Verletzt z.B. ein Datensatz den Foreign-Key-Constraint, wird die gesamte Transaktion zurückgerollt und man beginnt von Neuem. Über diesen Weg fehlt außerdem eine Aussage, welcher Datensatz die Verletzung verursachte.
- Nicht alle Datenformate sind kompatibel. In Abb. 13 und 14 ist zu sehen, wie z.B. Zahlen mit Nachkommastellen übernommen wurden.

```

COPY FROM WORSECK/PASSWORD@ODBC:RUEDIS CREATE RDB_BEHAN USING -
SELECT
  X_BEHAN_VID      , -
  X_BEHAN_TID      , -
  X_BEHAN_MID      , -
  T_BEHAN_USER     , -
  D_BEHAN_DATUM    , -
  X_BEHAN_BID      , -
  C_BEHAN_LETZTE   , -
  D_BEHAN_BEHAN    , -
  R_BEHAN_AUFWAND  *1000000 R_BEHAN_AUFWAND , -
  C_BEHAN_AUFWAN   , -
  R_BEHAN_WASSER  *1000000 R_BEHAN_WASSER , -
  C_BEHAN_WASSER   , -
  C_BEHAN_ANWTECH , -
FROM BEHAN_REL;
    
```

**Abb. 13:** Beispielscript zum Kopieren einer über ODBC eingebundenen RdB-Tabelle unter SQLPlus in eine ORACLE-Zwischentabelle

```

INSERT INTO RU_ANWENDUNG (
  VERSUCHS_NR,
  KENNR,
  NUTZER,
  DATUM,
  ANWENDUNG_NR,
  ANWENDUNG_AM,
  M_AUFWAND,
  M_AUFWAND_EINHEIT,
  W_AUFWAND,
  W_AUFWAND_EINHEIT,
  ANWENDUNGSTECHNIK)
SELECT
  X_BEHAN_VID *10 +X_BEHAN_TID,
  KONVERTIERE_MID(X_BEHAN_MID),
  RTRIM(T_BEHAN_USER),
  D_BEHAN_DATUM,
  X_BEHAN_BID,
  D_BEHAN_BEHAN,
  ROUND(R_BEHAN_AUFWAND/1000000,5),
  C_BEHAN_AUFWAN,
  ROUND(R_BEHAN_WASSER /1000000,5),
  C_BEHAN_WASSER,
  C_BEHAN_ANWTECH
FROM RDB_BEHAN
WHERE REPLACE(TO_CHAR(X_BEHAN_VID *10 +X_BEHAN_TID) ||KONVERTIERE_MID(X_BEHAN_MID),',','')
IN (SELECT REPLACE(TO_CHAR(B.VERSUCHS_NR)||B.KENNR,',','')
FROM RU_VERS_MITTEL B
WHERE B.VERSUCHS_NR=X_BEHAN_VID *10 +X_BEHAN_TID);
    
```

Abb.14: Beispielscript zum Generieren der ORACLE-Zieltabelle aus der in Abb. 13 geladenen Zwischentabelle

## 4 Management im laufenden Betrieb

### 4.1 Management – Datenbank

Der Oracle-Enterprise-Manager wird in der BBA genutzt, um INFOZUPF

- zu administrieren,
- die Datenbank zu überwachen, notwendige Maßnahmen abzuleiten und
- sie zu optimieren.

Die SQL-Statements mit nicht optimierten Zugriffen kann man herauszufinden (Abb. 15). Es werden Verbesserungsvorschläge unterbreitet (Abb. 16). Z.B. werden in INFOZUPF ca. 95% aller Anfragen an die DB aus dem Cache der Datenbank beantwortet.

The screenshot shows the Oracle SQL Analyze interface. The main window displays a table with the following columns: SQL-Text, Pufferlesezugriffe, Plattenlesezu..., and Ausführungen. The table lists various SQL queries and their corresponding statistics. Below the table, there is a 'Statistiken' section showing a specific SQL statement and its execution details.

SQL-Text	Pufferlesezugriffe	Plattenlesezu...	Ausführungen
SELECT COUNT(*) FROM BEMERKUNG B,WIRKSTOF...	66402	63727	153
SELECT COUNT(*) FROM BEWERTUNG B,WIRKSTOF...	57261	54378	153
SELECT TO_CHAR(COUNT(*) FROM V_NACHFORDE...	50243	29749	262
SELECT COUNT(*) FROM BEMERKUNG B,WIRKSTOF...	19740	16762	153
select distinct dok_deskriptor.deskriptor   :    substr(dok_ty...	23367	11101	152
SELECT /*+ NDEX(KODE_PK) *KODETEXT,SPEERE	1968033	6936	662697
SELECT COUNT(*) FROM BEWERTUNG B,WIRKSTOF...	54805	6687	153
SELECT COUNT(*) FROM BEWERTUNG B,WIRKSTOF...	9391	6172	153
SELECT COUNT(*) FROM BEMERKUNG B,WIRKSTOF...	63969	5928	153
SELECT MI.REFERENZ_KENNR,WG.WIRKNR FROM ...	16197	4699	22
SELECT TO_CHAR(COUNT(*) FROM BEMERKUNG W...	5698	4323	262
SELECT TO_CHAR(COUNT(*) FROM BEWERTUNG ...	5735	3644	262
SELECT NVL(MODUL_TITEL,?)MODUL_TITEL FROM...	19635	3480	1351
SELECT ROWID,KENNR,SPRACHE,KAPITEL,KAPITEL...	3775	3413	7
SELECT KODELISTE FROM KODELISTE_FELDNAME ...	3717292	3347	464692
select kodell :   substr(kodetext,1,40) a, kode b from db_A...	20016	2768	56
SELECT ROWID,PARTNER_KENNR,MISCHUNG_ART...	10236	2227	731

Statistiken SQL-Text

```

SELECT COUNT(*)
FROM bewertung b, wirkstoff_variante wv
WHERE b.substanz_nr = wv.substanz_nr
AND wirknr = :b1
    
```

Abb. 15: ORACLE-Tools zur Analyse der an die Datenbank abgesetzten SQL- Statements als Grundlage für eine Optimierung der Datenbank

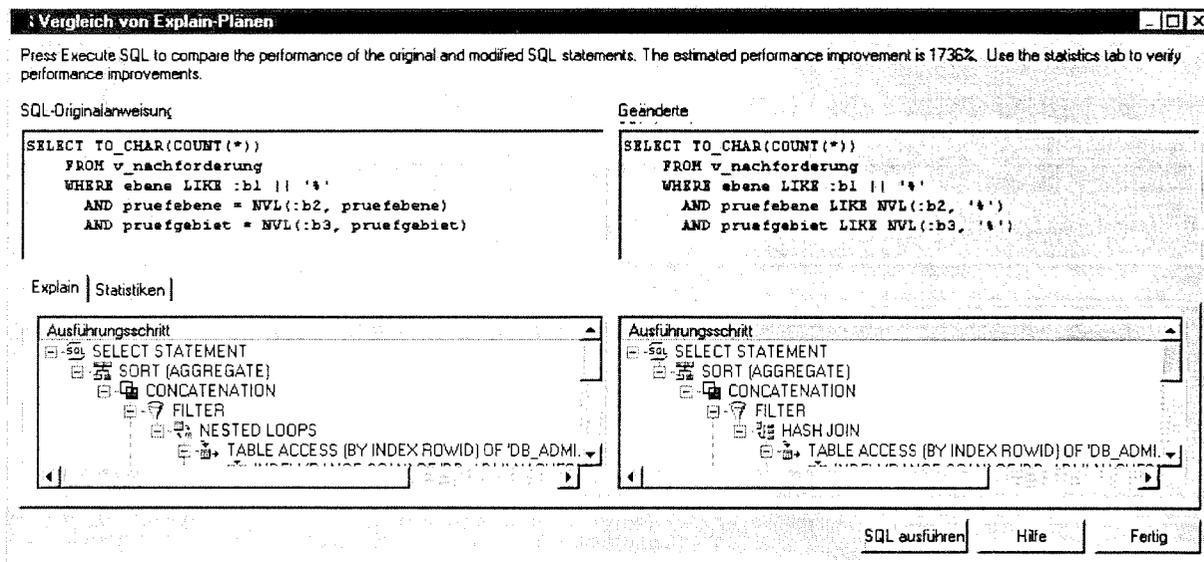


Abb. 16: Generierter SQL-Vorschlag zur Erhöhung der Performance einer Datenbankanfrage

## 4.2 Management – Kapazitätsengpässe

Irgendeine Stelle stellt den geschwindigkeitsbestimmenden Schritt in der Verarbeitung dar. Die Gründe für Engpässe sind vielschichtig. Einiges wird schon durch Kaufentscheidungen fest zementiert. Andere Kapazitätsengpässe kann man durch Umorganisation und geschickter Programmierung umgehen.

### DB-Server

- Beim Hardwarekauf bestimmen Anzahl der Prozessoren, Prozessorleistung, Hauptspeicher und IO-Kapazität des Servers maßgeblich die Performance.
- Wird der Server auch für andere Aufgaben eingesetzt, gibt es eine Konkurrenz zu anderen auf dem Server laufenden Nicht-DB Tasks (Abb. 17).
- Weitere Faktoren sind die Anzahl an gleichzeitigen DB-Prozessen (Abb. 18),
- der reservierter Hauptspeicher für die DB und
- die Verteilung der Zugriffe auf unterschiedliche Platten.

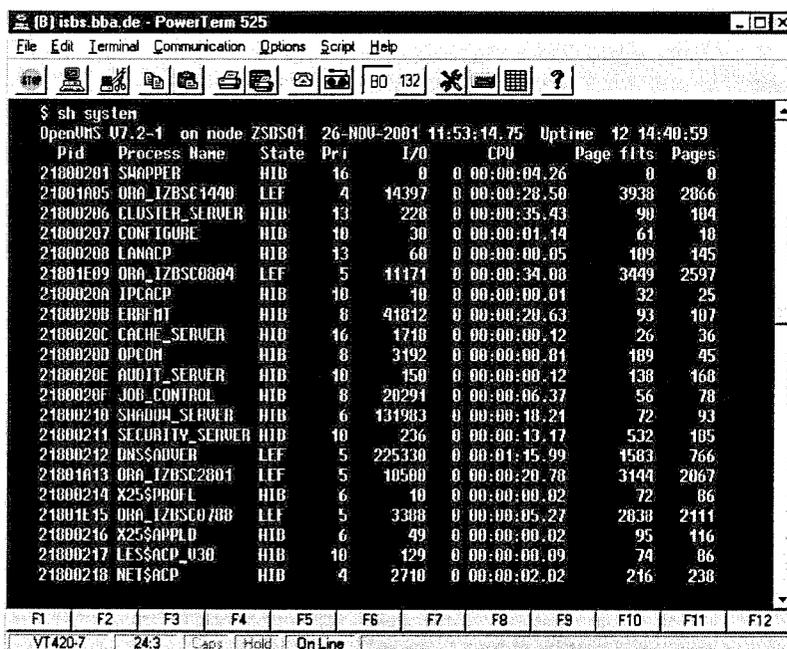


Abb. 17: Auf diesem Server laufen Datenbankprozesse und andere Dienste gleichzeitig

BENUTZERNA...	SID	OS-BENUT...	SESSION ...	BEFEHL	STATUS	MASCHINE	PROGRAM
DB_AI	63	db_	5971351	SELECT	ACTIVE	RZPC02	OBE60.EXE
WERN	24	wer	1345877	UNKNOWN	INACTIVE	ALPC05	
LITTM	29	littm	781787	UNKNOWN	INACTIVE	UFPC16	
PLEK	18	plef	777649	UNKNOWN	INACTIVE	KGPC19	
RICHT	81	rich	688054	UNKNOWN	INACTIVE	GPC27	
SONS	46	son	579557	UNKNOWN	INACTIVE	FCPC33	
BECKI	12	Bec	561844	UNKNOWN	INACTIVE	FCPC32	
FLEMI	14	flem	459385	UNKNOWN	INACTIVE	UFPC17	
ZAND	68	zan	451821	UNKNOWN	INACTIVE	IPPC21	
CONR	34	con	428285	UNKNOWN	INACTIVE	FBPC23	
HOHG	23	hoh	423990	SELECT	INACTIVE	FCPC03	
GILLE	35	gille	401171	UNKNOWN	INACTIVE	FCPC40	
HAAK	42	Ha:	382100	UNKNOWN	INACTIVE	IPPC07	
KULA	69	kul:	358938	UNKNOWN	INACTIVE	FBPC15	
KLOSI	56	klos	339554	UNKNOWN	INACTIVE	FBPC29	
SCHU	82	sch	311291	UNKNOWN	INACTIVE	FCPC27	
GILS_	11	gils	311080	UNKNOWN	INACTIVE	FBPC22	
HERL	52	herl	296922	UNKNOWN	INACTIVE	ALPC06	
SPINT	80	spir	293128	UNKNOWN	INACTIVE	ALPC04	
WOR:	95	wor	262745	UNKNOWN	INACTIVE	DVPC12	
LANGI	85	lang	250670	UNKNOWN	INACTIVE	FCPC26	
GEOR	48	geo	234664	UNKNOWN	ACTIVE	ALPC18	
CORD	32	corr	222718	UNKNOWN	INACTIVE	FCPC20	
ZELS	44	zels	174429	UNKNOWN	INACTIVE	FCPC21	
ZWEF	45	Zwr	169664	UNKNOWN	INACTIVE	UFPC100	
ILLME	36	ILLI	159026	UNKNOWN	INACTIVE	BBA_BS\AF...	
WITT	15	witt	156729	UNKNOWN	INACTIVE	ALPC14	
GLITS	75	glits	136922	UNKNOWN	ACTIVE	FCPC06	
HUET	83	hue	130262	UNKNOWN	INACTIVE	FCPC18	
GERS	17	ger:	118225	UNKNOWN	INACTIVE	ALPC07	
BUSC	30	bus	111861	UNKNOWN	INACTIVE	APC16	
BURO	57	bur:	110415	UNKNOWN	INACTIVE	FBPC51	
HAAK	41	Ha:	96407	UNKNOWN	INACTIVE	IPPC07	
STITO	31	STI	94595	UNKNOWN	INACTIVE	BBA_BS\AF...	
KRAL	60	kral	92745	UNKNOWN	INACTIVE	FBPC17	
STREI	43	stre	85593	UNKNOWN	INACTIVE	FBPC09	

Drücken Sie F1, um die Hilfe aufzurufen. Abgeglichen um: 10:03:12

Abb. 18: Die Datenbanknutzer konkurrieren um die Ressourcen auf dem Server

**Netzwerk**

- Zugriff auf die Programm-Shares über das lokale Netzwerk (400 Dateien mit 50 MB),
- Bei remote-Zugriff auf die DB: Auslastung der Standleitung (128 kBit) mit der Konkurrenz zu anderen Anwendungen (WWW, Mail).

**Client**

- Hauptspeicher (Nach eigener Erfahrung >=64 MB),
- Konkurrenz zu anderen Windows-Programmen.

Wird eine andere Softwarearchitektur eingesetzt, kommen weitere potentielle Kapazitätsengpässe hinzu.

## 5 Schlussbemerkung

Zum Zeitpunkt der Drucklegung dieses Artikels werden die 1995 getroffenen Grundsatzentscheidungen unter folgenden veränderten Rahmenbedingungen wiederum auf den Prüfstand gestellt:

- Reorganisation des Zulassungsverfahrens
- Online Einbindung der Einvernehmens-/Benehmens-Stellen
- Ankündigung der Firma ORACLE, die Client-Server Variante von Forms und Reports in naher Zukunft nicht mehr weiter zu entwickeln.

## Literatur

PFLANZENSCHUTZGESETZ (1998) Fassung vom 14. Mai 1998

(BGBl. I. S. 971, 1527, 3512), zuletzt geändert durch Artikel 4 des Gesetzes vom 6. August 2002

(BGBl. I. S. 3082,3087)

MICROSOFT (2001): Rich Text Format (RTF) Specification Version 1.7

Microsoft Technical Support 8/2001– Word 2002 RTF Specification



# TSN – bundeseinheitliches Werkzeug zum Krisenmanagement von Tierseuchen

Schröder, R., Klöß, D., Kroschewski, K., Staubach, C. und Micklich, A.

Bundesforschungsanstalt für Viruskrankheiten der Tiere

(ronald.schroeder@wus.bfav.de, detlef.kloess@wus.bfav.de, klaus.kroschewski@wus.bfav.de, christoph.staubach@wus.bfav.de und andreas.micklich@wus.bfav.de)

Seit 1995 nutzen alle Veterinärbehörden Deutschlands TSN als ein einheitliches elektronisches System zur Tierseuchenmeldung und -berichterstattung. Ursprünglich als reines Meldesystem unter MS-DOS konzipiert hat sich die Software in den letzten Jahren zu einem leistungsfähigen Werkzeug zum Krisenmanagement von Tierseuchen entwickelt.

Das System besteht aus zwei getrennten Komponenten. Im Zentrum steht mit der „Zentralen Tierseuchendatenbank“ ein Server, der zum einen alle Daten zu gemeldeten Tierseuchen in einer SQL-Datenbank speichert und zum anderen die komfortable Abfrage des Datenbestandes unter Nutzung von Internet-Technologien ermöglicht. Damit stehen einmal gemeldete Seuchenfälle allen angeschlossenen Veterinärbehörden praktisch in „Echtzeit“ zu Verfügung. Neben den üblichen Abfragen in Listenform kann die flexible kartografische Darstellung als Ausgabeform gewählt werden.

In den Veterinärämtern der Kreise und kreisfreien Städte, den für das Veterinärwesen zuständigen obersten Bundes- und Landesbehörden und Regierungspräsidien ist die Client-Komponente des TSN installiert. Diese erlaubt zum einen die Erfassung und Übertragung der tierseuchenrelevanten Daten. Zum anderen enthält sie umfangreiche Programmfunktionen zur Geografischen Informationsverarbeitung (GIS) sowie hoch auflösende digitale Landkarten, welche z.B. die exakte Bestimmung von Koordinaten (Georeferenzierung) bei Seuchenausbrüchen ermöglichen. Des weiteren erlaubt die Software die Verwaltung und Georeferenzierung komplexer Tierhalterbestände, um im Falle von Seuchenausbrüchen „auf Knopfdruck“ Restriktions- und Pufferzonen (Sperr- und Beobachtungsgebiete) bilden zu können. Dabei ist der Datenaustausch (Karten, Restriktionsgebiete) im Krisenfall zwischen involvierten Nachbarkreisen bzw. verschiedenen Behörden effizient möglich.

Ergänzende Elemente wie eine zentrale Adressverwaltung, umfangreiche Statistiken zu Tierseuchenausbrüchen, Hypertextsysteme mit Expertenwissen zu Tierseuchen sowie Schnittstellen zu anderen in Veterinärämtern verwendeten Programmsystemen runden das TSN als ein Informationssystem für alle Veterinärbehörden ab.

## 1 Vom Papier zu Bits und Bytes

Als 1992 mit der bundesweiten Einführung des EU-Projektes Animal Movement (ANIMO) zur Dokumentation des innergemeinschaftlichen Tierverkehrs jedes Kreisveterinäramt mit einem Personalcomputer und einem Modem ausgerüstet wurde, waren erstmals die technischen Voraussetzungen gegeben, innerhalb des Veterinärwesens „computergestützt“ kommunizieren zu können. Damit bestand die Möglichkeit durch Mitnutzung dieser Hardware, die bisherige „papiergestützte“ Berichterstattung zum Auftreten von anzeigepflichtigen Tierseuchen und meldepflichtigen Tierkrankheiten (TS/TK) durch ein zu schaffendes elektronisches Meldesystem zu ersetzen. Mit dieser Aufgabe wurde das Institut für Epidemiologie (IfE) der Bundesforschungsanstalt der Viruskrankheiten der Tiere (BFAV) betraut. Aufbauend auf einer Problemdokumentation wurde im Frühjahr 1993 mit der Entwicklung des Systems begonnen, das im Juli des gleichen Jahres unter dem Namen TierSeuchenNachrichten-System (TSN-System) erstmals Vertretern aller Bundesländer vorgestellt wurde. Nach einem Parallellauf mit dem bisherigen Meldesystem im Laufe des Jahres 1994 ist TSN seit 1995 das amtliche Meldesystem für anzeigepflichtige Tierseuchen.

## 2 Ein Server und viele Clients

Das System besteht aus zwei getrennten Komponenten. Im Zentrum steht mit der Zentralen Tierseuchendatenbank (ZTSDB) (Abb. 1) ein Server, der zum einen alle Daten zu TS/TK in einer Datenbank (INFORMIX) speichert und zum anderen die komfortable Abfrage des Datenbestandes unter Nutzung von Internettechnologie (IIS, CGI-Programme) ermöglicht. In den Veterinärämtern der Kreise und kreisfreien

Städte, den für das Veterinärwesen zuständigen obersten Bundes- und Landesbehörden, Regierungspräsidien sowie Untersuchungsämtern, Referenzzentren und Grenzkontrollstellen ist die Client-Komponente von TSN installiert. Mit Hilfe dieser Software werden Angaben zu TS/TK offline erfasst und per Knopfdruck zur ZTSDB übertragen. Entsprechend der Hoheit über die amtliche Feststellung von Seuchenfällen durch die örtlichen Veterinärbehörden ist die Änderung von Informationen in der ZTSDB ausschließlich dem jeweils zuständigen Kreisveterinäramt vorbehalten (schreibender Zugriff). Das Abrufen von Informationen aus der ZTSDB (lesender Zugriff) ist dem gesamten Kreis der ca. 500 genannten Nutzer möglich.

### 3 TSN wird grafisch oder der Schritt zu Windows

Bei den ersten Versionen der lokalen Komponente des TSN handelte es sich um eine Software, die für das seinerzeit dominierende PC-Betriebssystem MS-DOS entwickelt wurde. Nach 6 Jahren des Einsatzes von insgesamt drei TSN-Versionen für MS-DOS, ist seit Dezember 2000 die Variante für Windows verfügbar. Die Neuprogrammierung des TSN für eine grafische Oberfläche wurde genutzt, um Funktionalität und Bedienerfreundlichkeit des Systems erheblich zu erweitern.

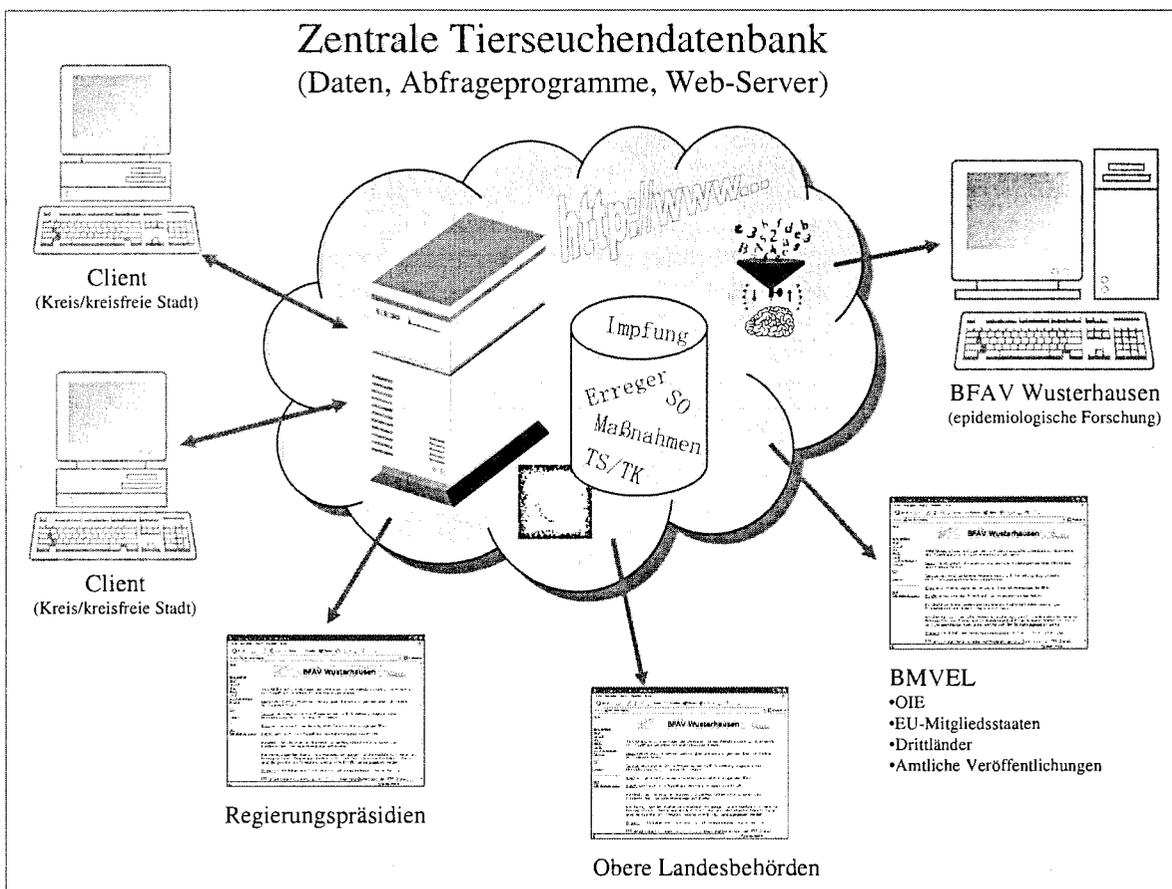


Abb. 1: Zentrale Tierseuchendatenbank

Die Client-Software ist durch folgende Merkmale gekennzeichnet:

- Kompatibel zu allen 32-bit-Windows-Versionen,
- Kommunikation über die DFÜ-Schnittstelle von Windows (Modem, ISDN), über Intranet (ISDN-Router im lokalen Netzwerk) oder direkt über das öffentliche Internet,
- Stammdatengesteuerte Erfassung von Informationen zu TS/TK,
- Automatische Synchronisation von Bewegungsdaten (Meldungen zu TS/TK) und Stammdaten (Steuerdaten zur Datenerfassung, administrative Grenzen zur kartografischen Darstellung usw.),
- Verwaltung und Auswertung von Informationen zu Tierhaltern mit Schnittstellen zu anderen in Veterinärämtern genutzten Systemen,

- GIS-Funktionalitäten zum Krisenmanagement,
- Umfangreiche Online-Dokumentation auf HTML-Basis,
- Einbindung einer Wissenssammlung zu diagnostischen Methoden auf HTML-Basis.

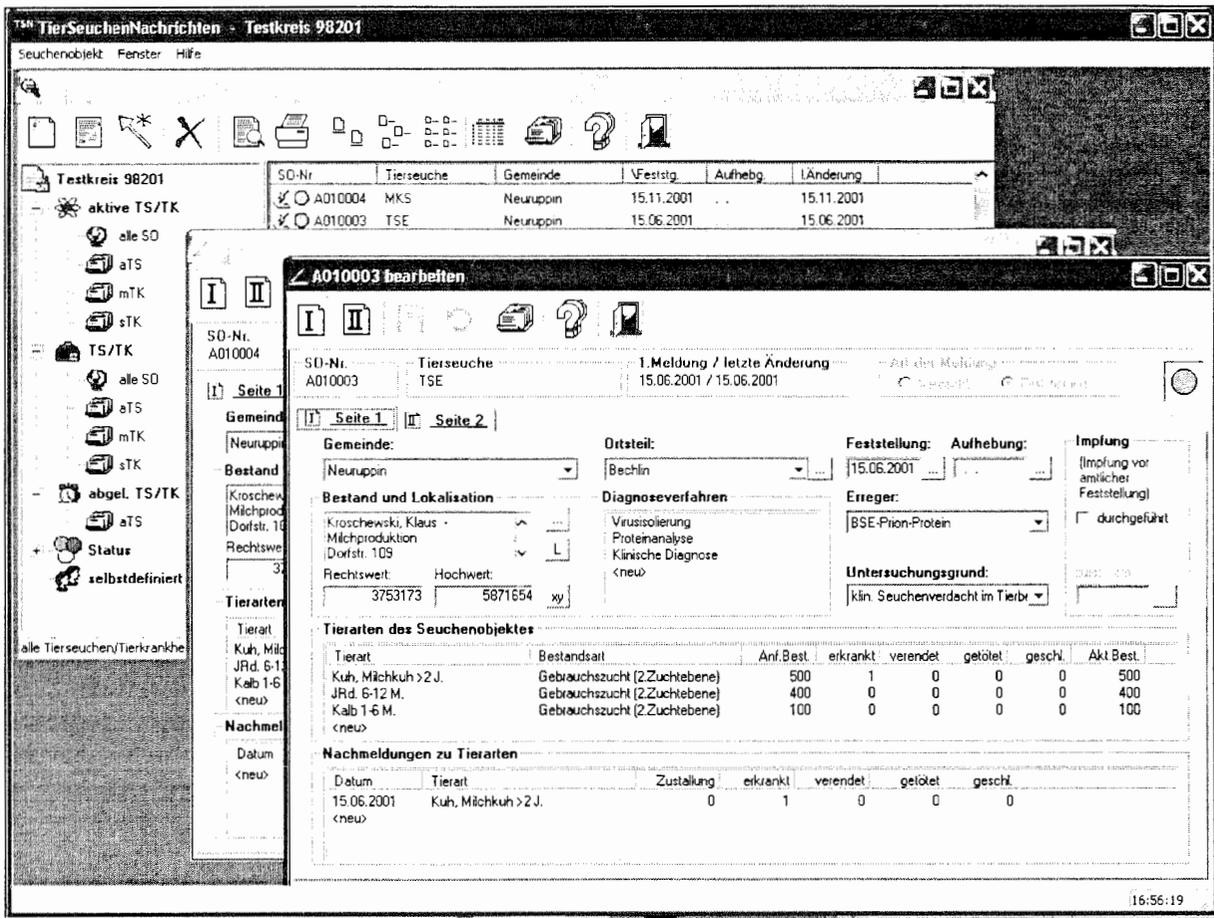


Abb. 2: Bildschirmmaske des Seuchenobjekt-Editors

#### 4 TSN-Online – „Informationen in Echtzeit“

Greifen die Clients der früheren DOS-Versionen von TSN noch in einem zeilenorientierten Terminalfenster auf die Informationen in der ZTSDB zu, geschieht das seit der Version 2.0 grafisch unter Nutzung eines Internetbrowsers. Die Verbindung zwischen Client und Server kann mit dem integrierten DFÜ-Netzwerk von Windows über Modem oder ISDN-Karte durch Einwahl in das Intranet der BFAV hergestellt werden oder neuerdings direkt über das öffentliche Internet erfolgen. Damit ist theoretisch vom jedem internetfähigen PC weltweit der Zugriff auf die ZTSDB möglich. Wengleich der Zugriff jederzeit und praktisch von überall her möglich ist, bleibt er jedoch einzig einer geschlossenen TSN-Benutzergruppe (alle Veterinärbehörden, die über Nutzernamen und Passwort für TSN-Zugang verfügen) vorbehalten.

Jede Änderung in der ZTSDB steht zeitgleich allen Clients zur Verfügung. Damit ist die Aktualität der ZTSDB als einheitliches Informationssystem für alle Ebenen der Veterinärbehörden ausschließlich von der zeitnahen Meldung des Auftretens von TS/TK durch den zuständigen Amtstierarzt abhängig. Alle Abfragen der ZTSDB erfolgen über intuitive Webseiten, die über die Spezifizierung zeitlicher, regionaler und inhaltlicher Parameter (z.B. einzelne TS/TK) schnell zu den gewünschten Detailinformationen führen. Neben der reinen Listendarstellung können die Darstellungen auch wahlweise kartografisch erfolgen. Dazu wurde ein so genannter Karten-Server unter Nutzung von COM-Technologie auf Basis von ESRI MapObjects entwickelt.

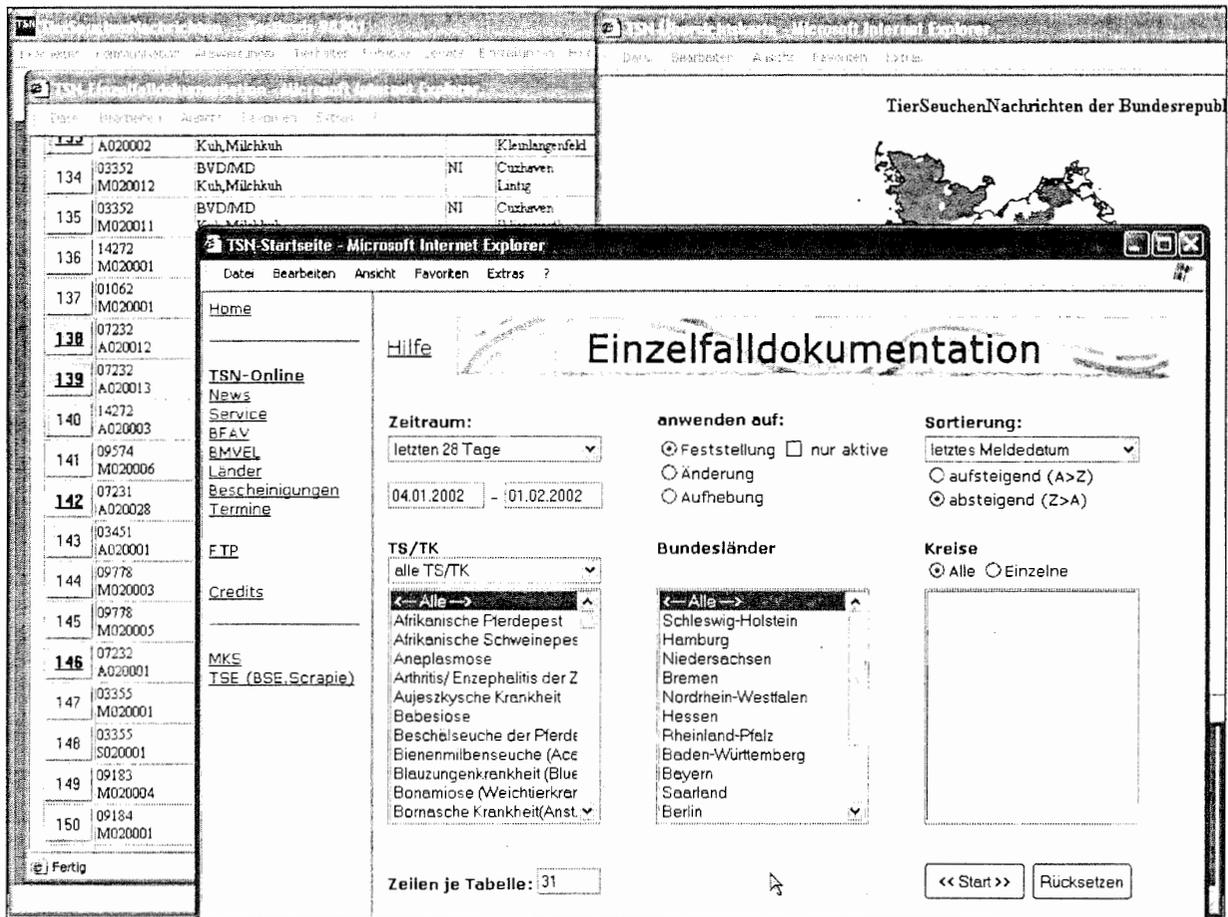


Abb. 3: Bildschirmmasken von TSN-Online

Die Funktionalitäten des Online-Teils von TSN sind jedoch nicht nur auf die reine Abfrage der ZTSDB beschränkt. Zusätzlich ist eine Adressverwaltung aller TSN-Nutzer respektive aller Veterinärbehörden Deutschlands seit Januar 2002 verfügbar, die von jedem Nutzer (für seine eigenen Angaben) online gepflegt wird und über verschiedene Suchfunktionen den Zugriff auf gesuchte Personen und Einrichtungen einfach ermöglicht. Über diese Adressverwaltung kann jeder Nutzer einstellen, ob er automatisiert per Email über den Neuausbruch wählbarer TS/TK im Regierungsbezirk, Bundesland oder Deutschland informiert werden möchte.

Folgende weitere Optionen stehen der Benutzergruppe zur Verfügung:

- Eingescannte Statistiken (z.B. monatliche Tierseuchenberichte),
- Informationen und Fragebögen zu ausgewählten Tierseuchen,
- Veterinärbescheinigungen,
- Downloadbereich für neue Programmversionen und Karten,
- Diverse Homepages (BFAV, ausgewählte Informationen der BMVEL-Homepage).

## 5 TSN wird kartografisch oder GIS wird integriert

Etwa 90 Prozent aller weltweit erfassten Datenbestände verfügen über einen konkreten räumlichen Bezug, der im Allgemeinen über die Angabe von Adressen hergestellt wird. Die Erfahrungen mit den ersten TSN-Versionen haben gezeigt, dass im Rahmen der Tierseuchenbekämpfung und für die Anwendung mathematischer Modelle und von Methoden räumlicher Simulation die Angaben von Gemeinden, Orten oder Strassen keine praktikable Genauigkeit erreichen. Insbesondere ist die räumliche Zuordnung von Seuchenfällen bei Wildtieren damit nicht möglich. Für eine effiziente Tierseuchenbekämpfung der Behörden (z.B. Bildung von Restriktionsgebieten) und wissenschaftliche Betrachtungen ist bei hochkontagiösen Tierseuchen eine exakte

Zuordnung auf Basis von Koordinaten erforderlich. Diesen Forderungen sollte bei der Entwicklung der Windowsversion des TSN Rechnung getragen werden.

Um Koordinaten zu ermitteln und entsprechende räumliche Abfragen und kartografische Darstellungen zu generieren ist i.d.R. eine GIS-Software (Geografisches Informationssystem) notwendig. Die anfänglichen Hoffnungen des TSN-Entwicklerteams, dass in den Veterinärämtern geeignete GIS-Systeme und geeignetes Kartenmaterial bereits verfügbar sind oder deren Beschaffung geplant ist, erwiesen sich als trügerisch. Einige wenige Bundesländer wie Sachsen oder Brandenburg bzw. einzelne Kreise mit besonders hohen Bestandsdichten wie Cloppenburg in Niedersachsen hatten bereits Projekte dieser Art in Angriff genommen. Dabei wurden zur Georeferenzierung und kartografischen Darstellung ausnahmslos externe GIS-Applikationen wie ArcView oder PolyGIS verwendet. Im überwiegenden Teil Deutschlands waren insbesondere aus Kostengründen keine ähnlichen Aktivitäten geplant. Um eine deutschlandweite Verfügbarkeit der räumlichen Zuordnung von Seuchenfällen und Beständen zu ermöglichen, galt es, kurzfristig die entsprechende Funktionalität in TSN zu integrieren, was unter Nutzung der COM-Komponente MapObjects LT erfolgte.

Damit wurden bereits in die erste Windowsversion von TSN alle GIS-Funktionen aufgenommen, die dem grundsätzlichen „GIS-Bedarf“ eines Kreisveterinäramtes entsprechen:

1. komfortable Georeferenzierung aller Seuchenfälle und Bestände (Tierhaltungen) des Kreises,
2. kartografische Darstellung aller bzw. ausgewählter Tierhalterinformationen,
3. Bildung von Sperr- und Beobachtungsgebieten (Radien),
4. Bildung frei definierbarer Sperrgebiete anhand natürlicher Grenzen (Polygone) und
5. Kompatibilität zu Nachbarkreisen (z.B. Austausch von Polygonen für Sperrgebiete, die Grenzen von Kreisen bzw. Bundesländern überschreiten).

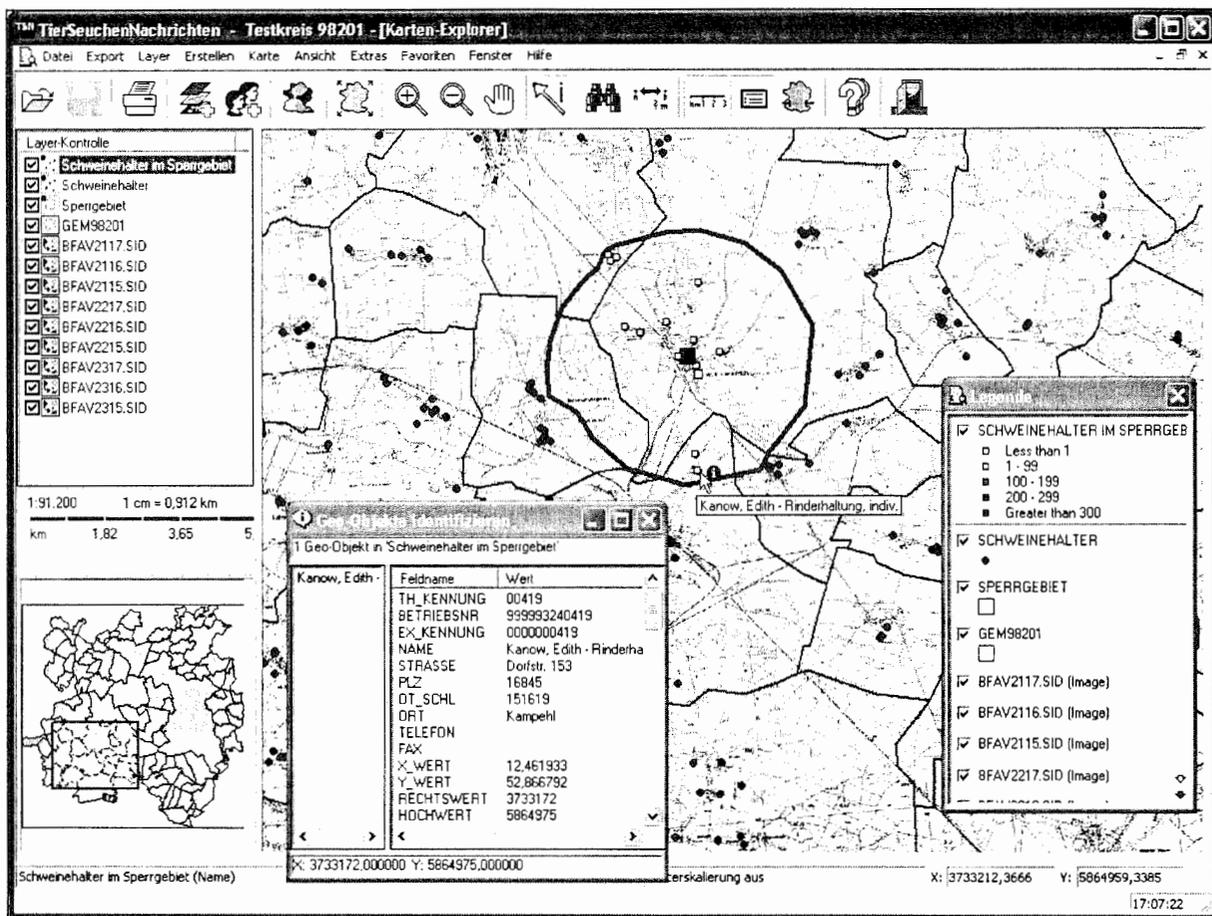


Abb. 4: Bildschirmmaske des Karten-Explorer

Da die Integration der Funktionen per COM-Technologie direkt in das Programm erfolgte (keine externe Software mehr nötig), ist die Bedienung intuitiv und ohne weiteres Spezialwissen möglich. Die Software kann in den Ämtern beliebig oft installiert werden, kostspielige externe GIS-Software wird i.d.R. nicht mehr benötigt. Um die Anwendung der kartografischen Funktionen sofort verfügbar zu machen, liefert und installiert TSN die erforderlichen Vektor- und Rasterkarten (administrative Grenzen und topografische Hintergrundkarten) in Übereinstimmung mit den Lizenzbedingungen der Landesvermessungsämter als Eigentümer der Rechte an diesen Karten mit. Als Kartengrundlage wird ein deutschlandweit einheitlich projiziertes Kartenwerk (Gauß-Krüger-System, Potsdam Datum, 3. Meridianstreifen) verwendet, so dass bei Anwendung von TSN die Kompatibilität aller georeferenzierten Daten automatisch gegeben ist, was für die Anwendung in Krisenzentren von großem Vorteil ist.

## **6 Krisenmanagement inklusive**

Waren die Kapazitäten für die Verwaltung von Informationen zu Beständen und zur räumlichen Abfrage bzw. kartografischen Darstellung bisher auf die Belange von Kreisämtern abgestimmt, erweitert die Version 2.2 diese Fähigkeiten auf regionale Krisenzentren. Im Bedarfsfall können die georeferenzierten Bestandsdaten mehrerer Kreise eingelesen werden. Ein intuitives und komplexes Darstellungs- und Abfragewerkzeug, der Karten-Explorer, erlaubt die flexible Gestaltung und Einfärbung von Karten. Geografische Selektions- und Suchfunktionen, Export- und Druckfunktionen sind verfügbar. Fragestellungen wie „Wie viele Schweinebestände befinden sich im 3-km-Radius von Schlachthöfen?“ oder „Welche Bestände befinden sich innerhalb sich überlappender Sperrgebiete?“ können mit wenigen Handgriffen beantwortet werden. Sperrbezirke und Beobachtungsgebiete, die über Kreisgrenzen hinweggehen stellen kein Problem mehr dar. Die einzigen Voraussetzungen für die Arbeit sind das Vorhandensein der entsprechenden Hintergrundkarten (topografische Karten) und georeferenzierter Informationen (Tierhalter, Betriebe usw.). Der Karten-Explorer ist zudem in der Lage per Download gespeicherte Abfragen der ZTSDB (z.B. aktuelle Situation der Klassischen Schweinepest in Deutschland) als Karte zur individuellen Gestaltung bzw. Druckausgabe weiter zu verarbeiten.

## **7 Stand und Ausblick**

Die Windowsversion 2.0 von TSN ist derzeit nahezu flächendeckend installiert. Seitens der BFAV wurden in der Zeit von September 2001 bis Januar 2003 12 Schulungsveranstaltungen mit insgesamt ca. 600 Teilnehmern durchgeführt.

Die Rasterkarten (DTK25) als Grundlage für die Georeferenzierung wurden mit erfolgter Genehmigung der jeweiligen Landesvermessungsämter an die Kreise aller Bundesländer der Bundesrepublik mit Ausnahme der „Stadtstaaten“ Berlin und Hamburg weitergegeben. Die umfassende Georeferenzierung von Beständen und Seuchenfällen hat in vielen Bundesländern begonnen.

Die seit August 2002 verfügbare Version 2.2 erlaubt die effektive Nutzung des TSN als Werkzeug für übergreifende Krisenzentren. Die weitere Entwicklung der Client-Komponente wird den Weg zum Werkzeug für das Krisenmanagement konsequent weiter verfolgen.

Durch die direkte Kommunikation auch über das öffentliche Internet ist die Verfügbarkeit der ZTSDB als Informationssystem für alle Veterinärbehörden praktisch uneingeschränkt gegeben. Eine Weiterentwicklung des Servers in Richtung Auskunftssystem für relevante Rechtsvorschriften, Desinfektionsrichtlinien, internationale Tierseuchensituation -wie auf den besagten TSN-Schulungen von zahlreichen Teilnehmern immer wieder angeregt- wäre bei entsprechender externer Unterstützung der TSN-Arbeitsgruppe wünschenswert. Die technische Plattform zum Angebot von jeglichen Informationen für die geschlossene Benutzergruppe Veterinärbehörden ist vorhanden.

Die einheitliche Erfassung und Georeferenzierung von Tierhaltern mit der lokalen Komponente von TSN in den Kreisen würde den Aufbau einer zentralen georeferenzierten Tierhalterdatei (sofern erwünscht) möglich machen. Im Jahre 2002 wurde im IfE mit dem Aufbau einer geografischen Datenbank mit detaillierten

Informationen auf Grundlage der ATKIS-Daten unter Nutzung der Software ArcSDE (Spatial Database Engine ) begonnen. Weitere Informationen stehen z.B. durch den zentralen ANIMO-Server zur Verfügung, der ebenfalls vom IfE betreut wird.

Die Verknüpfung der genannten Datenbestände und die Möglichkeit der Anwendung mathematischer Modelle und räumlicher Simulationen auf Grundlage dieser Datenbestände, die ggf. per Online-Abfrage verfügbar gemacht werden können, lassen der Fantasie für die weitere Entwicklung des TSN in den nächsten Jahren viel Raum.

## Abkürzungsverzeichnis

ANIMO	Animal Movement
ATKIS	Amtliches Topografisches Kartografisches Informationssystem der Bundesrepublik Deutschland
BFAV	Bundesforschungsanstalt für Viruskrankheiten der Tiere
BMVEL	Bundesministerium für Verbraucherschutz, Ernährung und Landwirtschaft
CGI	Common Gateway Interface
COM	Component Object Model, einheitliche Schnittstelle zur Integration und Fernsteuerung von Softwarekomponenten
DFÜ	Datenfernübertragung
DTK25	Deutsche Topografische Karte 1:25.000
EDV	Elektronische Datenverarbeitung
ESRI	Environmental Systems Research Institute, Redlands, USA
EU	Europäische Union
GIS	Geografisches Informationssystem
HTML	Hypertext Markup Language
IIS	Internet Information Server
IfE	Institut für Epidemiologie
ISDN	Integrated Services Digital Network
MS-DOS	Microsoft Disc Operation System
PC	Personalcomputer
SDE	Spatial Database Engine, Software zur Speicherung von geografischen Informationen in SQL-Datenbanken und zum effizienten Zugriff auf diese
TS/TK	Tierseuche/ Tierkrankheit
TSN	Tierseuchennachrichten
ZTSDB	Zentrale Tierseuchendatenbank



# Datenbankmanagement mit raumbezogenen Inhalten

Staubach, C.<sup>1</sup>, Klöß, D.<sup>1</sup>, Schröder, R.<sup>1</sup> und Busse, W.<sup>2</sup>

<sup>1</sup> Bundesforschungsanstalt für Viruskrankheiten der Tiere

(staubach@wus.bfav.de, detlef.kloess@wus.bfav.de, ronald.schroeder@wus.bfav.de)

<sup>2</sup> greenlab geoinformatics GmbH

(wb@greenlab.de)

Im Laufe der Entwicklung von geografischen Informationssystemen wurden verschiedene Konzepte der Modellierung von raumbezogenen Daten entworfen. Dabei kann die reale Welt grundsätzlich in gerasterter Form oder als Vektorgrafik modelliert werden. Rasterdaten folgen bei nicht-proprietären Formaten, wie z.B. TIFF, JPEG, am wenigsten dem Prinzip von relationalen Datenbankstrukturen. Aber schon die auf GIS-Anwendungen spezialisierten Rasterformate und erst recht Vektordaten nutzen die Vorteile einer effizienten Datenspeicherung in einer relationalen Datenbankstruktur. Die Zusammenführung von Geometrie- und Sachdaten zur effizienten Speicherung und Analyse von Daten in einer SQL-Datenbank ist möglich.

## 1 Einleitung

Eine schlüssige Definition von raumbezogenen Daten findet sich bei BURROUGH und MCDONNELL (1998): „Alle Datenbestände zu denen sich durch ein Koordinatensystem oder Adressen ein Raumbezug auf der Erdoberfläche herstellen lässt, sind raumbezogene Daten“. Überlegungen zu Dimension und Umfang eines möglichen Datenbestandes führen schnell zu der Erkenntnis, dass die meisten Daten auf dieser Erde in irgendeiner Form einen Raumbezug haben. Selbst gemäßigte Anforderungen im Bezug auf die geforderte Detailgenauigkeit und Größe des jeweiligen Gebietes können schnell zu enormen Datenmengen führen und werfen schnell die Frage nach einem effizienten Datenbankmanagement auf. Die zentrale Rolle spielen hierbei Geographische Informationssysteme (GIS), welche die Aufgabe der Erfassung, Speicherung, Überprüfung, Integration, Manipulation, Analyse und Darstellung von raumbezogenen Daten erfüllen (CHORLEY, 1987). Zur effizienten Speicherung und Auswertung digitaler Daten mit Raumbezug benutzen fast alle GIS das Prinzip der relationalen Datenbankstruktur jedoch implementiert in Form von speziellen Dateiformaten (LONGLEY u.a., 1999). In diesem Beitrag werden an Hand von Datenformaten der Firma ESRI (Redlands, CA, USA) Prinzipien der Datenbankstruktur und die speziellen Anforderungen an das Datenmanagement von geographischen Daten erläutert (ANONYMOUS, 1990; ANONYMOUS, 1998; LONGLEY u.a., 2001).

## 2 Geographische Informationssysteme (GIS)

Geographische Informationssysteme dienen der Visualisierung, Abfrage und Analyse von raumbezogenen Daten. Karten werden seit einigen Tausend Jahren zur Darstellung und Analyse von geografischen Informationen verwendet. GIS setzen diese alte Tradition mit neuen Methoden fort, indem kartografische Daten mit bestimmten Charakteristika (z.B. Krankheit, geologische oder ökologische Parameter) verknüpft werden. Seit den 90er Jahren fand im Zuge des rasanten Fortschritts von Hard- und Software auch eine Weiterentwicklung der GIS auf einen mehr anwenderbezogenen Einsatz hin statt (GOODCHILD u.a., 1992).

Unter dem weit gefassten und viel verwendeten Begriff der GIS werden in der Leistung sehr verschiedene Programme zusammen gefasst. Dies sind zum einen, Systeme mit denen geographische Daten erstellt, verändert und analysiert werden können (z.B. ArcInfo, ArcView, Atlas, PolyGIS, MapInfo, etc.). Zum anderen Desktop Mapping Software (Kartierungs-Programme, wie z.B. EpiMap, Regiograph) in denen Daten mit Raumbezug dargestellt werden ohne jedoch deren Geometrie verändern zu können. In dieser Kategorie existieren auch frei verfügbare GIS-Tools zur Darstellung und einfachen Abfrage von räumlichen Daten, wie z.B. der ArcExplorer oder Proviewer, die jedoch an spezielle Datenformate gebunden sind und über keine Import- oder Exportmöglichkeiten verfügen. Die dritte Kategorie umfasst Datenbanken (z.B. MS Access, Informix, Oracle, MS SQL-Server), die neben der Information über Objekte, wie Gehöfte oder Fuchsbauten, die dazugehörigen geographischen Punktkoordinaten erfassen können. Betrachtet man die

technische Entwicklung der letzten Jahren, so ist eine zunehmende Annäherung der Systeme zu beobachten. Einerseits integrieren die Desktop Mapping Systeme immer häufiger Funktionalitäten von GIS im engeren Sinne, und andererseits werden deren kartografische Ausgaben ausgereifter. Jedoch sollte immer berücksichtigt werden, dass die kartographischen Produkte eines GIS in erster Linie der Analyse von Daten dienen. Im Vordergrund steht die korrekte Visualisierung und weniger die Ästhetik.

Die Sachdaten bzw. Attributtabelle zu den raumbezogenen Objekten werden in der Regel in relationalen Datenbanksystemen vorgehalten, die mit Standardsoftware verändert werden kann (z.B. dBASE, Access, INFO). Erweiterungen von modernen Client-/Server-basierten SQL-Datenbanken (z.B. Informix, Oracle, MS SQL-Server) erlauben sogar die effiziente Speicherung von Vektor- und Rasterdaten sowie deren Attributtabelle in einer gemeinsamen Datenbankstruktur und den Zugriff auf diese Daten über das Intra- und/oder Internet.

### 3 GIS-Datenmodelle

Raumbezogene Daten, welche den Schlüssel zur Nutzung von GIS darstellen, stehen in immer größerem Umfang zur Verfügung. Vor allem die Detailliertheit der zur Verfügung stehenden Daten und damit die Dateigröße sowie die Sachinformationen nahmen in den letzten Jahren überdurchschnittlich zu. Als Beispiel dafür wäre das Amtlich-Topografisch-Kartographische Informationssystem (ATKIS) der Bundesrepublik Deutschland zu erwähnen, wo die Topografie der Landschaft mit hoher Genauigkeit als digitales Geländemodell modelliert wird und in einer einheitlichen Datenstruktur abgelegt ist.

Es werden zwei Kategorien von Daten benötigt, um eine Karte zu entwerfen, die Geometrie- und die Sachdaten. Unter Geometriedaten werden alle Angaben zum räumlichen Bezug verstanden, also alle Koordinaten über Punkte, Linien und die Ausdehnung von Flächen im dargestellten Gebiet. Die Sachdaten enthalten die Informationen und sind mit den Geometriedaten verknüpft und können den geometrischen Grundeinheiten Punkt, Linie oder Fläche räumlich zugeordnet werden. Daraus lassen sich unterschiedliche Methoden kartographischer Darstellung ableiten. Natürlich sind diese Methoden beliebig zu kombinieren, genauso wie unterschiedliche Daten in einer Karte dargestellt werden können.

Die digitale Abbildung der realen Welt lässt sich in einem GIS grundsätzlich in zwei verschiedenen Datenmodellen realisieren, als Raster- oder Vektordaten. Dabei wird die Landschaft entweder in ein Gitter von gleich großen quadratischen Zellen zerlegt oder beschrieben durch Punkte, Linien und/oder Polygone je nach geeigneter Landschaftskategorie (Abb. 1).

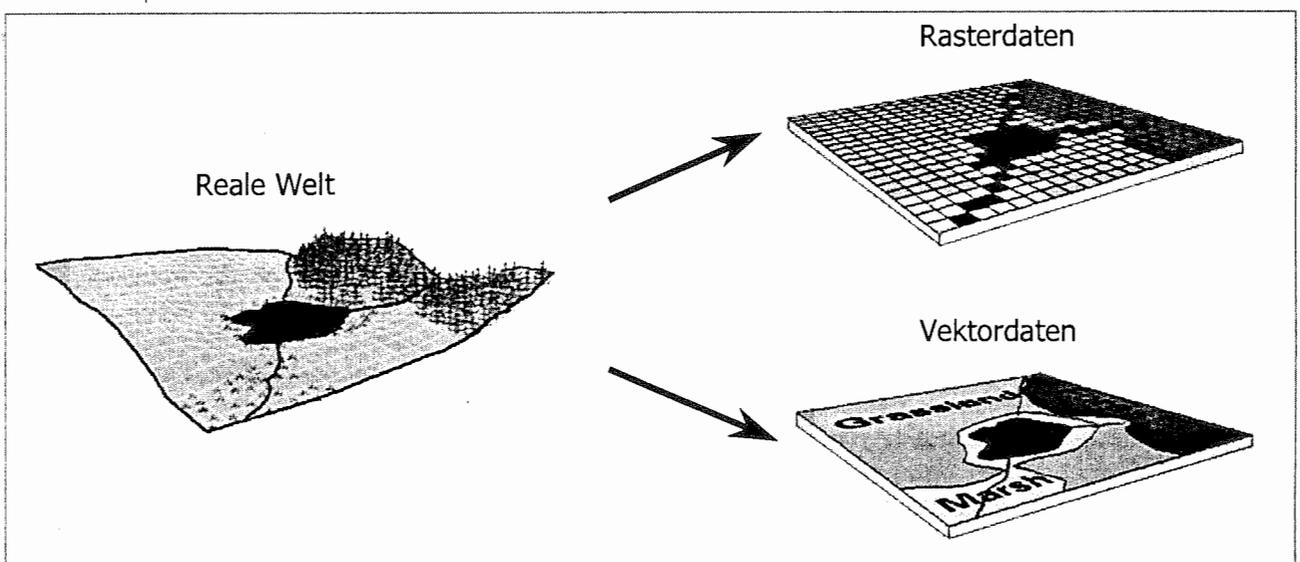
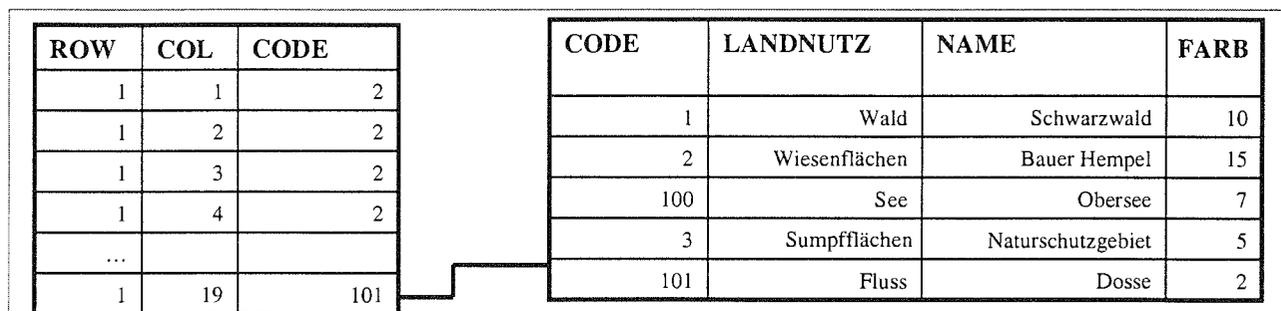


Abb. 1: Grundsätzliche Datenmodelle in GIS (modifiziert mit freundlicher Genehmigung von Philipp Hurvitz)

### 3.1 Rasterdaten

Rasterdaten sind eine gleichmäßige Anordnung von rechteckigen Zellen, angeordnet in einer Matrix nach Zeilen und Spalten. Bei proprietären Rasterdatenformaten welche ausschliesslich in GIS Verwendung finden, wie z.B. das GRID-Format der Firma ESRI, können die Zellen mit einer Zeile in einer Sachdatentabelle verknüpft werden (Abb. 2).



**Abb. 2:** Grundsätzlicher Aufbau von proprietären Rasterdatenformaten (modifiziert nach ANONYMOUS (1990))

Da eine Bilddatei ebenfalls als ein Raster mit einer Zellgröße entsprechend der Pixelgröße des jeweiligen Bildes interpretiert werden kann, können z.B. TIFF-, JPEG- oder BSQ-Bilddateien ebenfalls nach der Georeferenzierung in einem GIS verwendet werden. Dabei wird der Raumbezug durch Bestimmung der Pixelgröße und exakten Position des Bildausschnittes in der realen Welt hergestellt. Die Farb- bzw. Pixelwerte des Bildes stellen dabei die Sachdaten des Rasters da und können zum einen mit unterschiedlichen Grau- bzw. Farbwerten für die Visualisierung belegt werden, zum anderen können auch die Werte selbst für Analysen herangezogen werden (z.B. die Berechnung von Vegetationsindizes auf Grundlage der Pixelwerte verschiedener Spektralbänder von Satellitenaufnahmen; LILLESAND und KIEFER, 1994; STAUBACH u.a., 2001).

Rasterdaten zeichnen sich durch eine einfache Datenstruktur aus, die eine effiziente Analyse von kontinuierlich verteilten räumlichen Daten ermöglicht. Die Datenstruktur bedingt, dass der Zugriff auf einzelne Rasterzellen einfach ist, jedoch der Zugriff auf Objekte, wie z.B. Wald oder Wiese, schwierig ist. Da die Detailliertheit der Daten ausschließlich von der Zellgröße in der realen Welt abhängt und alle Rasterzellen belegt sein müssen, ob sie nun Informationen enthalten oder nicht, ergibt sich ein hoher Speicherplatzbedarf im Vergleich zu Vektordaten (LIEBIG, 1999).

Rasterdaten dienen v.a. der Darstellung von kontinuierlichen Daten, wie z.B. Höhen-, Wetter-, Satellitendaten, Luftbildaufnahmen, der Verteilung des Stickstoffs im Boden, sowie zur Hinterlegung von Vektordaten oder der Produktion von Atlanten meistens in Form von digitalisierten oder gescannten topographischen Karten. Beispiele finden sich in Form der digitalen topografischen Karte 1:25.000 (DTK25) in TSN (siehe dieser Band) und bei LILLESAND und KIEFER (1994), BURROUGH und MCDONNELL (1998), LONGLEY u.a. (1999 bzw. 2001).

### 3.2 Vektordaten

Vektordaten stellen die Geometrie eines Objektes in der realen Welt entweder als Punkte, Linien oder Polygone in einem kartesischen Koordinatensystem dar. Vom Basiselement Punkt ausgehend wird für jedes Objekt ein einzelnes x- und y-Koordinatenpaar in der Geometrietabelle abgelegt und bei punktförmigen Objekten, wie z.B. Bestände, Bäume oder Fuchspositionen, durch eine eindeutigen Schlüssel mit der Sachdatentabelle verbunden. Mit den Punkten lassen sich auch hierarchische Objekte konstruieren, die aus Linien oder Polygonen bestehen. Eine Linie ist dann die Verbindung von zwei oder mehreren x-y-Koordinatenpaaren, welche über den gleichen Schlüssel in der Geometrietabelle dem jeweiligen Linienobjekt zugeordnet werden. Linien dienen z.B. der Darstellung von Strassen oder Gewässern und besitzen als Attribut die Länge der Linie für jedes Objekt. Polygone sind zweidimensionale Objekte

bestehend aus vier oder mehreren geordneten und verbundenen x-y-Koordinaten. Das erste und letzte x-y-Koordinatenpaar sind gleich und schließen eine Fläche ein. Auch hier dient der gemeinsame Schlüssel aller Koordinaten-paare der Identifikation eines Polygon-objektes. Mit Hilfe von Polygonen werden flächenhafte Objekte wie z.B. Siedlungs-flächen, Seen, Wälder oder administrative Gebiete dargestellt. Als zusätzliche Eigenschaftsattribute stehen dem Anwender für jedes Polygon Umfang und Flächengröße zur Verfügung, die von einem GIS bei der Bildung der Polygone berechnet werden können (Abb. 3).

Polygone können prinzipiell auf zwei unterschiedlichen Wegen modelliert werden. Zum einen auf der Basis von Punktdaten, die mit Hilfe von eindeutigen Schlüsseln zu Polygonen zusammengefügt werden (Abb. 4a). Zum anderen über ein hierarchisches Modell bei dem zunächst die Seiten als Linien modelliert werden und dann zu einem Polygon zusammengefügt werden (Abb. 4b).

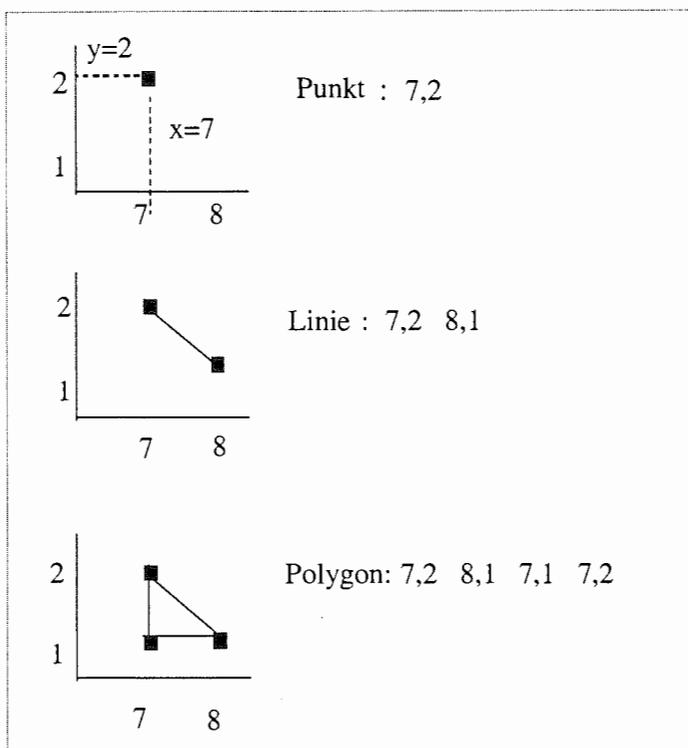


Abb. 3: Grundsätzlicher Aufbau von Vektordatenformaten (modifiziert nach ANONYMOUS (1990))

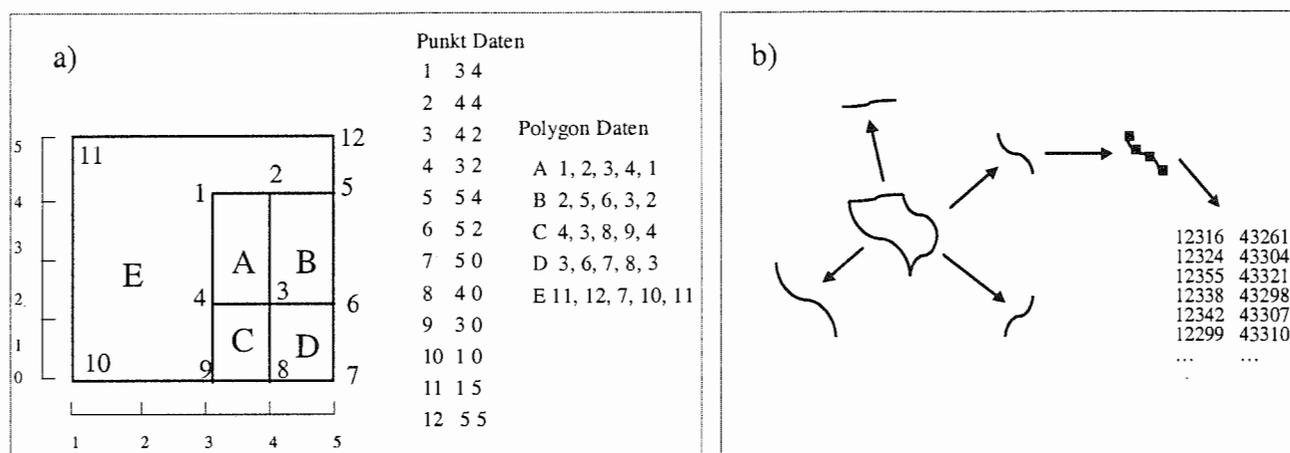


Abb. 4: Möglichkeiten der Modellierung von Polygonen (4a modifiziert nach ANONYMOUS (1990))

Alle Vektordaten verfügen zusätzlich auch über Attributdaten, die das Aussehen der jeweiligen Vektorgrafik beschreiben, wie z.B. die Größe und Farbe der Punkte, Stärke und Farbe der Linien oder Füllfarbe und -muster der Polygone. Die Beschreibung der räumlichen Nachbarschaft zwischen den einzelnen Vektorobjekten wird als Topologie bezeichnet. Zum Beispiel ist es für räumliche Abfragen essentiell zu wissen, welche Polygone direkt benachbart oder durch eine bestimmte Linie voneinander getrennt sind. Die Information über die Topologie kann, wie z.B. bei dem ArcInfo-Format der Firma ESRI, als Indexdateien für jedes Objekt ständig vorliegen oder kann, wie z.B. beim Shape-Format, erst „on-the-fly“ während der Abfrage durch das GIS erzeugt werden.

Vektordaten ermöglichen dem Nutzer sehr einfach auf Objekte zuzugreifen und räumliche Selektionen durchzuführen. Trotz der komplexen Datenstruktur benötigen Vektordaten durch die effektive Datenverwaltung nur wenig Speicherplatz im Vergleich zu Rasterdaten. Dagegen ist die Analyse von

kontinuierlich verteilten räumlichen Daten nur mit komplexen Auswerteprogrammen möglich (LIEBIG, 1999).

Beispiele für Vektordaten finden sich in Form der digitalisierten Tierbestände, administrativen Grenzen oder Restriktionsgebiete in TSN (siehe dieser Band) und bei LILLESAND und KIEFER (1994), BURROUGH und MCDONNELL (1998), LONGLEY u.a. (1999 bzw. 2001).

#### 4 Integrierte raumbezogene Datenbankmodelle

Die Zusammenführung von Geometrie- und Sachdaten in einem relationalen Datenbanksystem hat vor allem bei der Nutzung von raumbezogenen Datenquellen durch mehrere GIS-Anwender große Vorteile. Zum einen wird der Datenbestand an einer zentralen Stelle mit gleichem Stand für alle Nutzer vorgehalten, zum anderen können mehrere Anwender auch mit unterschiedlichen GIS-Lösungen gleichzeitig über ein TCP/IP-Netzwerk auf die Daten zugreifen. Dabei kann man zusätzlich die Funktionalitäten moderner SQL-Datenbanken nutzen wie z.B. Skalierbarkeit, Transaktionen oder die individuelle Vergabe von Lese- und/oder Schreibrechten für die unterschiedlichen Kartenschichten für jeden Nutzer.

Die Geometriedaten, d.h. die Koordinaten, werden nicht sichtbar für den Nutzer in Datenbankfeldern (i.d.R. BLOB-Feldern) abgelegt und über einen eindeutigen Schlüssel mit der für den Anwender sichtbaren Datenbanktabelle verbunden. Eine Kartenschicht besteht dann aus einer Kollektion von Tabellen (LONGLEY u.a. 1999 bzw. 2001). Ein räumlicher Index erlaubt die schnelle Darstellung und effiziente räumliche Analyse auch bei sehr großen Datenmengen. Bei diesem Verfahren wird ein Gitternetz über die gesamte Ausdehnung der Kartenschicht gelegt, und in einer räumlichen Indextabelle werden für die maximale Ausdehnung eines jeden Geoobjektes die Gitterzellen abgelegt, welche durch die Objekte berührt oder geschnitten werden (Abb. 5).

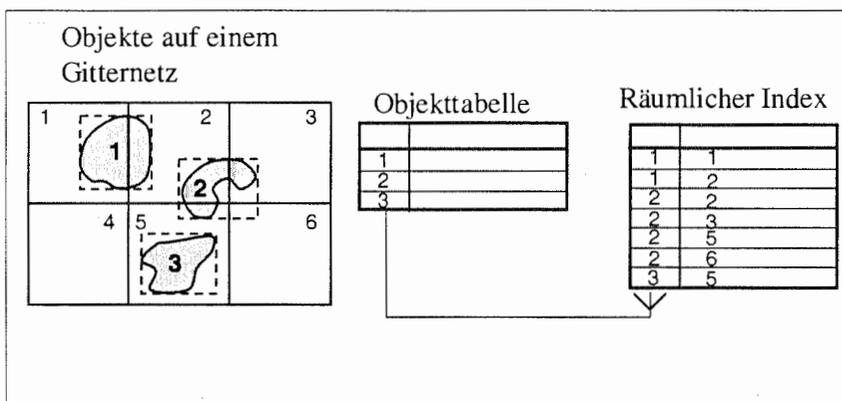


Abb. 5: Raumbezogenes Indizieren (modifiziert nach LONGLEY u.a., 2001)

Nicht verschwiegen werden darf, dass die Zusammenführung von verschiedenen Kartenschichten aus mehreren Datenquellen, sei es in einem integrierten raumbezogenen Datenbankmodell oder als Einzelplatzlösung, Hürden birgt, wie z.B. unterschiedliche Geodatenformate und Projektionen, so dass man aufwendige Konvertierungen in Kauf nehmen muss. Da es eine fast unendlich große Anzahl von Möglichkeiten gibt, die dreidimensionale Erde auf einer Ebene (topografischen Karte) mathematisch abzubilden, entwickelten sich unterschiedlichste Projektionsverfahren und Koordinatensysteme. In Folge dessen und durch kommerziell nicht einheitlich erhältliche Software entstanden parallel unterschiedlichste Datenformate, die nur durch GIS im engeren Sinne ineinander umwandelbar sind. Deshalb ist es notwendig, bei der Überlagerung von mehreren unterschiedlichen Datenquellen sehr vorsichtig zu sein und immer die Projektion, das Koordinatensystem und die Auflösung der einzelnen Originaldaten zu berücksichtigen.

## Literatur

- ANONYMOUS (1990): Understanding GIS – The ARC/INFO Method  
Environmental Research Institute, Redlands
- ANONYMOUS (1998): ESRI Shapefile Technical Description - An ESRI White Paper  
Environmental Research Institute, Redlands
- BURROUGH, P.A. und MCDONNELL, R.A. (1998): Principles of Geographical Information Systems  
Oxford University Press, Oxford
- CHORLEY, R.J. (1987): Handling Geographic Information  
The Report of the Committee of Inquiry Chaired by Lord Chorley, HMSO, London
- GOODCHILD, M.F., HAINING, R. und WISE, S. (1992): Integrating GIS and spatial data analysis: problems and possibilities  
International Journal of Geographical Information Systems, 6, 407-423.
- LIEBIG, W. (1999): Desktop-GIS mit ArcView GIS  
Wichmann Verlag, Heidelberg
- LILLESAND, T.M. und KIEFER, R.W. (1994): Remote sensing and image interpretation  
John Wiley & Sons Inc., New York
- LONGLEY, P.A., GOODCHILD, M.F., MAGUIRE, D.J. und RHIND, D.W. (1999): Geographical Information Systems  
Vol. 1 und 2, John Wiley & Sons Inc., New York
- LONGLEY, P.A., GOODCHILD, M.F., MAGUIRE, D.J. und RHIND, D.W. (2001): Geographical Information Systems and Science  
John Wiley & Sons, New York
- STAUBACH, C., THULKE, H.-H., TACKMANN, K., HUGH-JONES, H. CONRATHS, F.J. (2001): Geographical information system-aided analysis of factors potentially influencing the spatial distribution of *Echinococcus multilocularis* infections of foxes  
The American Journal of Tropical Medicine and Hygiene 65(6), 943-948.

# Datenmanagement in der BAZ am Beispiel der Kohl-Datenbank

Kecke, S.

Bundesanstalt für Züchtungsforschung an Kulturpflanzen

(s.kecke@bafz.de)

Ausgehend vom zunehmenden Bedarf an Werkzeugen zur arbeitsgruppen- und institutsübergreifenden Handhabung von immer größeren Datenmengen wurde 2001 in der BAZ eine Arbeitsgruppe „Datenmanagement“ ins Leben gerufen. Vertreter aller Institute legten in der Diskussion die Grundlagen für den Aufbau eines dezentralen Datenmanagementsystems mit einem zentralen Kern am Hauptstandort. Das in der Arbeitsgruppe EDV der BAZ seit 1999 bearbeitete Projekt zum Aufbau einer Datenbanklösung zur Speicherung und Bearbeitung von Daten der Gattung *brassica* bildet eine der Keimzellen des künftigen Gesamtsystems. In diesem Beitrag werden die Grundlagen, das Konzept, verwendete Software sowie die bisher realisierte Softwarelösung erläutert und demonstriert.

## 1 Notwendigkeit eines Datenmanagement-Konzeptes

In zunehmendem Maße werden in den Instituten der BAZ Lösungen für die Handhabung, Bearbeitung, Speicherung und Auswertung von immer größeren Datenmengen benötigt. Einerseits ist ein starkes quantitatives Wachstum der Datenmengen zu beobachten, andererseits müssen die Daten aufgrund der nicht mitwachsenden personellen Kapazitäten immer effektiver verwaltet und ausgewertet werden. Das erfordert in zunehmendem Maße den Einsatz von Werkzeugen der Informatik und unter diesen zuallererst die Nutzung zeitgemäßer Datenbanktechnologien.

Waren die ersten anfallenden Aufgabenstellungen noch mit lokalen Lösungen von Mitarbeitern der Arbeitsgruppe EDV (AG EDV) zu lösen, so zeigte sich bald, dass dieser Lösungsansatz zu kurz greift:

Der Zugriff auf die Daten über das Netz ist nicht bzw. nur eingeschränkt möglich.

Die Daten müssen lokal an vielen Stellen gesichert werden.

Es besteht die potenzielle Gefahr, dass Nutzer mit eigenen Mitteln die Daten manipulieren und damit die Konsistenz gefährden.

Viele Daten werden redundant vorgehalten.

Kein Nutzer hat Zugriff auf Daten anderer, wodurch möglicherweise neue Zusammenhänge hergestellt und die eigenen Daten in größeren Zusammenhängen betrachtet werden könnten.

Eine Projektion von Daten im Intra- oder Internet ist auf direktem Wege nicht möglich.

Methoden der verteilten Datenbearbeitung über das Internet können nicht angewendet werden.

Diese und weitere Einschränkungen führten dazu, dass man sich in der AG EDV mit der Technologie der relationalen Datenbank-Managementsysteme (RDBMS) zu beschäftigen begann. Im Interesse eines einfachen und schnellen Zugangs und ohne erst die Hürden einer aufwändigen Finanzierung eines kommerziellen Systems wie z. B. Oracle nehmen zu müssen, entschied man sich für eine durchgängige Open-Source-Lösung.

So wird seit ca. 3 Jahren erfolgreich mit MySQL auf Linux-Plattform gearbeitet. Einen Überblick über die wichtigsten Projekte der letzten Jahre gibt Abb. 1.

Wie aus der Darstellung der Abbildung bereits zu erkennen ist, bilden ein Teil der Daten eine Schnittmenge. Diese muss zur Zeit bei Bedarf noch redundant vorgehalten werden. Bis auf die Dokumentation der Genbank in Braunschweig, die bereits als Oracle-Datenbanklösung vorlag, entstanden die genannten Anwendungen auf der Grundlage lokaler Datenbankformate, wie dBase oder Paradox., bzw. von MySQL. Als Werkzeuge für die Nutzerschnittstellen (Frontends) kommen Delphi (Borland) bzw. PHP zum Einsatz.

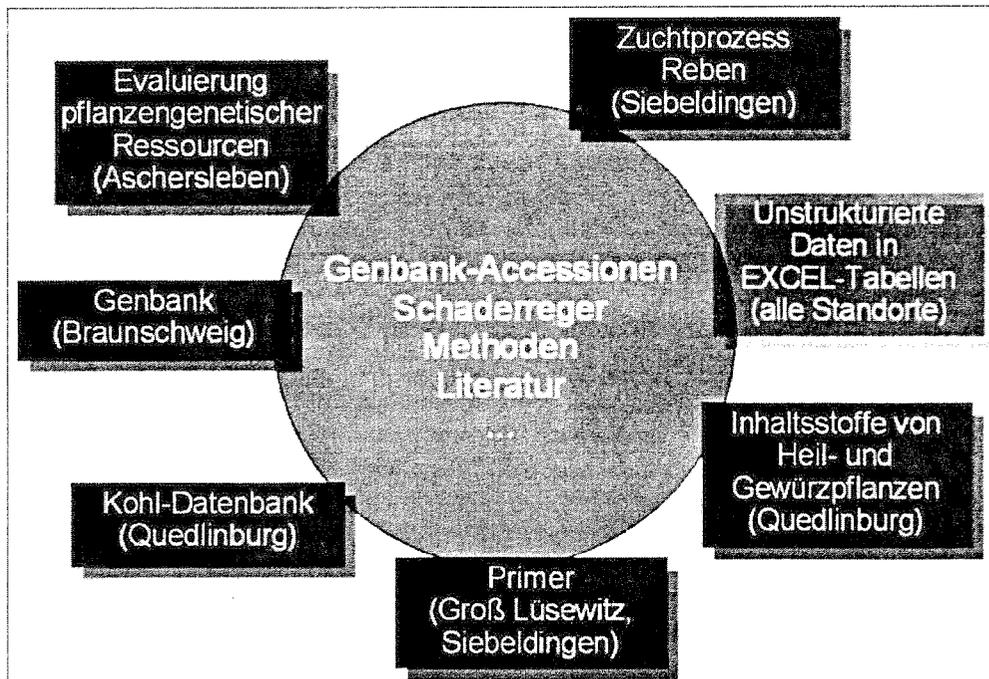


Abb. 1: Bestandsaufnahme

Ein großer Teil der Forschungsdaten liegt in den BAZ-Instituten jedoch nach wie vor in Form von Excel- oder Textdateien vor und ist somit dem allgemeinen Zugriff und vor allem der späteren Weiterverarbeitung und Auswertung nur schwer oder gar nicht mehr zugänglich.

Zusammenfassend kann also eingeschätzt werden, dass die Forschungsdaten in den BAZ-Instituten in sehr heterogener Form vorliegen und eine direkte Weiterverarbeitung und Auswertung kaum möglich ist. Im Laufe der Zeit sind mehrere Insellösungen entwickelt worden. Mit der Nutzung des RDBMS MySQL entstanden erste Lösungsansätze mit der Möglichkeit der zentralen Datenhaltung und der verteilten Nutzung über das Medium Internet.

Etwa zu diesem Zeitpunkt fanden sich Vertreter aller Institute auf Initiative des Leiters der BAZ zusammen, um über das Thema „Datenmanagement“ in der BAZ zu beraten und gründeten eine entsprechende BAZ-Arbeitsgruppe.

Hauptaufgabe dieser ist es nun, auf der Grundlage einer umfassenden Bestandsanalyse und der bereits vorhandenen Insellösungen eine Strategie für ein zukünftiges Datenmanagement in der BAZ zu entwickeln und auf den Weg zu bringen.

Am Beispiel der so genannten Kohl-Datenbank soll hier anhand einer der bereits vorliegenden Einzellösungen ein möglicher Ansatz erläutert werden.

## 2 Die Kohl-Datenbank

### 2.1 Modell

Im Laufe der Projektlaufzeit wurden interessante Erfahrungen mit dem Prozess der Beschreibung der Daten, aber vor allem der Beschreibung und Formalisierung von Arbeitsabläufen gesammelt. Es hat sich sehr schnell herausgestellt, dass der Begriff „Datenmanagement“ nur einen Teil des Problems beleuchtet. Besser ist wahrscheinlich der Begriff „Wissensmanagement“, wie er in STAAB (2002) verwendet wird, denn worum es im Umgang mit den Forschungsdaten der BAZ-Institute eigentlich geht, ist vielmehr das Zurverfügungstellen von Wissen, als nur das Ablegen von Daten.

Das dem Projekt zugrunde liegende Problem bestand darin, dass der Informationsfluss über die Pflanzen, deren Zustand, Standort und die Ergebnisse verschiedener an ihnen vorgenommener Untersuchungen dokumentiert werden sollte. Darüber hinaus sollten alle Versuchsdaten zentral abgelegt werden, so dass sie

später weiteren Auswertungen zur Verfügung stehen können. Abb. 2 zeigt in einer vereinfachten Darstellung die Projektion des realen Lebensprozesse von Pflanzen der Gattung *brassica* auf ein datenbankbasiertes Work-Flow-Modell.

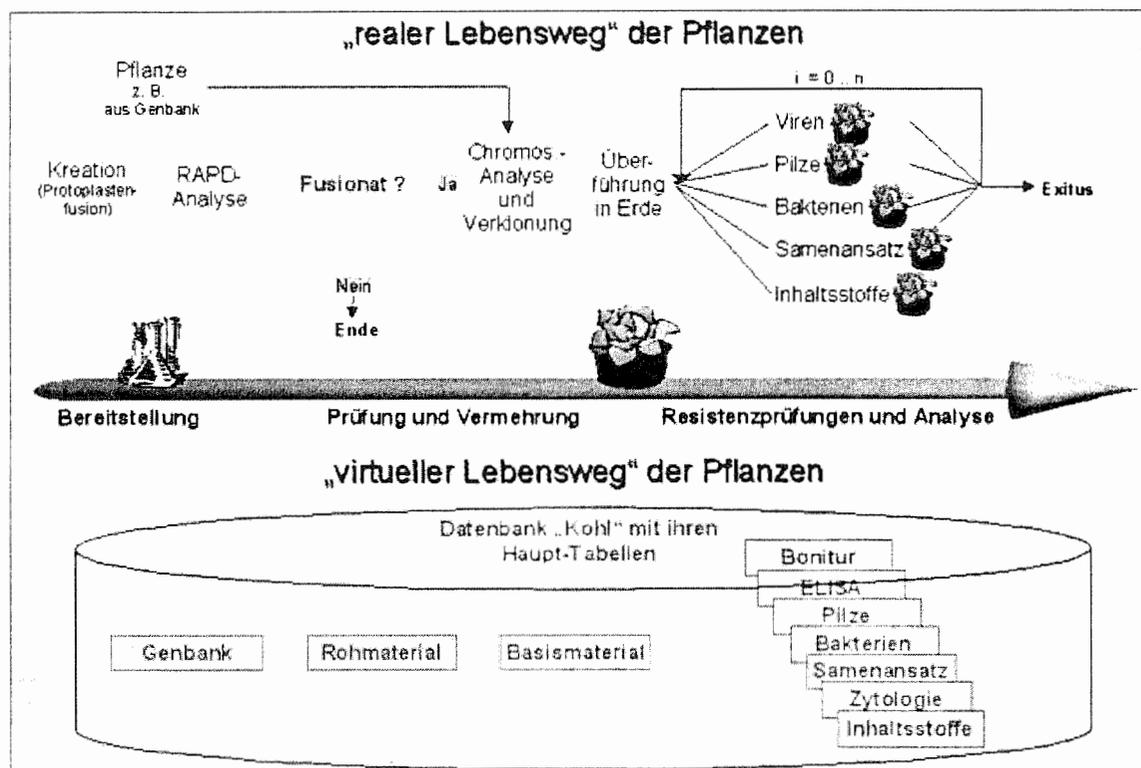


Abb. 2: Modell

Das Modell besteht aus dem Datenschema, also der Beschreibung der zu verwendenden Datenbanktabellen und ihrer Relationen, sowie aus der Beschreibung der Abläufe bzw. Informationsweitergabe. Ausgehend von der Modellbeschreibung wurden zur Realisierung das relationale Datenmodell (hier MySQL) und eine grafische Benutzeroberfläche für Windows-PC (hier Delphi der Firma Borland) ausgewählt.

Das vereinfachte Datenmodell zeigt die Abb. 3.

Grundlegender Gedanke des Projektes ist die Handhabung der während des gesamten Lebens der einzelnen Pflanzen anfallenden Daten über mehrere Arbeitsgruppen mit verschiedenen Aufgaben hinweg. Jede Arbeitsgruppe soll ihre eigene Sicht auf die Daten und die Möglichkeit der Eingabe und Bearbeitung ihrer Daten bekommen. Darüber hinaus können alle auf zusammenfassende Auswertungen zugreifen.

Die Phase der Erarbeitung des (vorläufigen) Daten- und Ablaufmodells wurde in mehrere Etappen eingeteilt. Nach Beratungen mit jeder einzelnen beteiligten Arbeitsgruppe, in denen die für sie relevanten Details besprochen wurden, fand ein erstes gemeinsames Kolloquium statt, in dem das Gesamtmodell in einer ersten Version vorgestellt wurde. Dieser Prozess wiederholte sich zum Teil, weil es sich zeigte, dass die Einigung auf gemeinsame Begriffe, Bezeichnungen, Namen, Verantwortlichkeiten, Abläufe u. a. schwieriger als erwartet war.

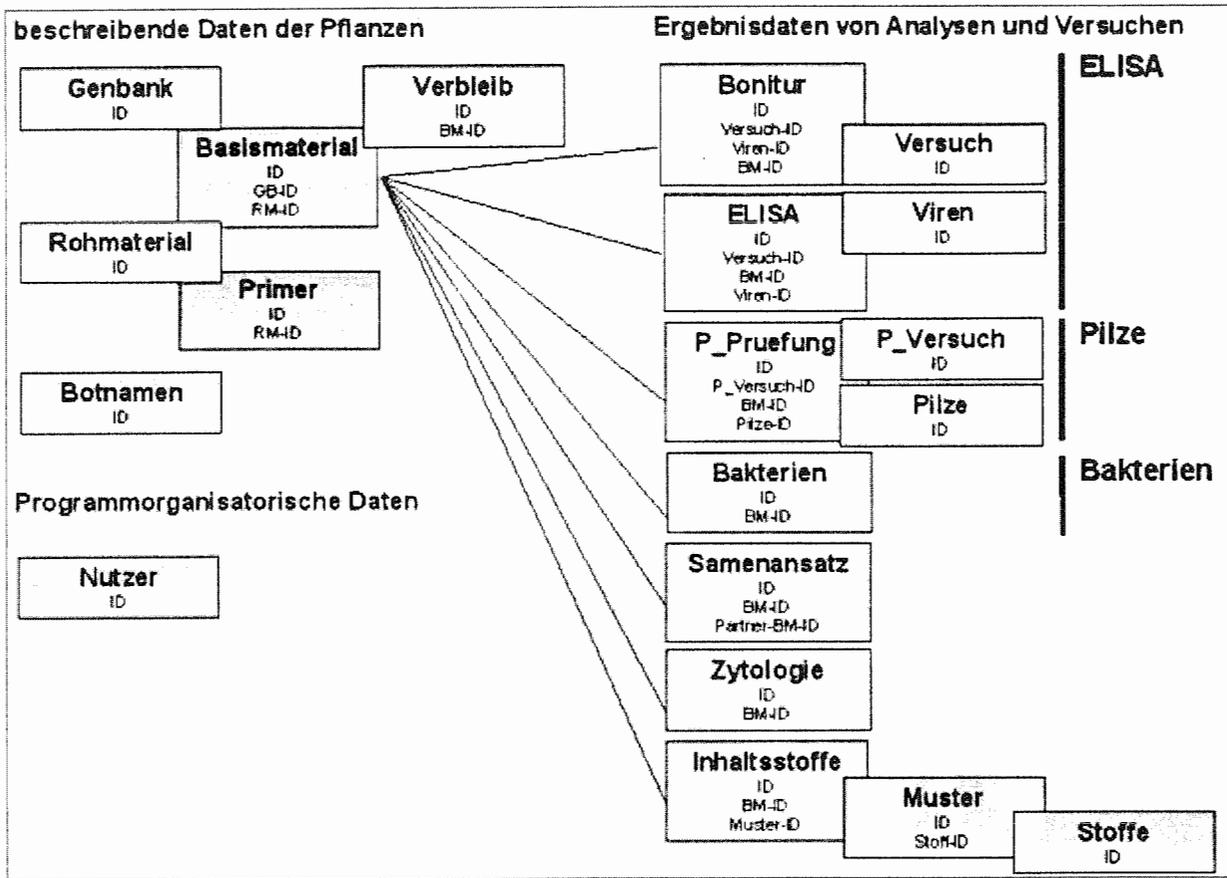


Abb. 3: Vereinfachtes Datenmodell

## 2.2 Programm

Nachdem das Modell in seinen wesentlichen Teilen aufgestellt war, begann die Implementierung der Datenbank und des Programms, mit welchem die Nutzer die Daten eingeben und bearbeiten können. Nach mehrjährigen praktischen Erfahrungen wurde auf die Programmiersprache Delphi zurückgegriffen, welche sehr gut geeignet ist, über ODBC-Teiber auf MySQL-Datenbanken im Netz zuzugreifen und auch komplexe Funktionen zu realisieren.

Für Nutzer aus allen Arbeitsgruppen wurde ein Programm konzipiert, welches sich nach der namentlichen Anmeldung eines Nutzers mit den spezifischen Funktionen für die jeweilige Arbeitsgruppe zeigt. Für komplexe Auswertungen der Daten können später weitere Programme entwickelt werden.

Entsprechend der bisher in das Projekt integrierten Arbeitsgruppen wurden 6 Nutzergruppen definiert:

- **Basismaterial**  
(Verwaltung, Vermehrung und Weitergabe von meist geklonten Einzelpflanzen an die Arbeitsgruppen)
- **Rohmaterial**  
(Erstellung von Rohmaterial mit verschiedenen Methoden und Erzeugung von Basismaterialpflanzen, sogenannter Klonteile)
- **Primer**  
(RAPD-Analyse von Rohmaterialpflanzen)
- **Viren**  
(Virusresistenzprüfungen mit verschiedenen Methoden)
- **Pilze**  
(Pilzresistenzprüfungen mit verschiedenen Methoden)

- Inhaltsstoffe  
(Analyse bestimmter Inhaltsstoffe von Pflanzen oder Pflanzenteilen)

Jeder Nutzernamen ist genau einer Gruppe zugeordnet.

## 2.2.1 Gruppe 1 – Basismaterial

Abb. 4 zeigt den Startbildschirm für die Gruppe 1. Die Startbildschirme der anderen Gruppen sind äquivalent, lediglich die zur Verfügung stehenden Funktionen und die Farben der Tabellenfelder variieren. Mit der Gestaltung des Startbildschirms wird versucht, den Ablauf und die zur Verfügung stehenden Möglichkeiten zu verdeutlichen. Die Farbe der Felder, durch welche die wichtigsten Datenbanktabellen symbolisiert werden, gibt Auskunft darüber, was der Nutzer für Zugriffsmöglichkeiten hat. Grün steht für Lesen und Schreiben, Gelb für Lesen, Rot für gesperrt. Ebenfalls nutzerabhängig werden durch Schaltfelder mit roten Pfeilen wichtige Bearbeitungsfunktionen hervorgehoben. Die Bearbeitung der Daten erfolgt entweder über das Menü, Klick auf die roten Pfeile oder die Kontextmenüs, die den Tabellen zugeordnet sind.

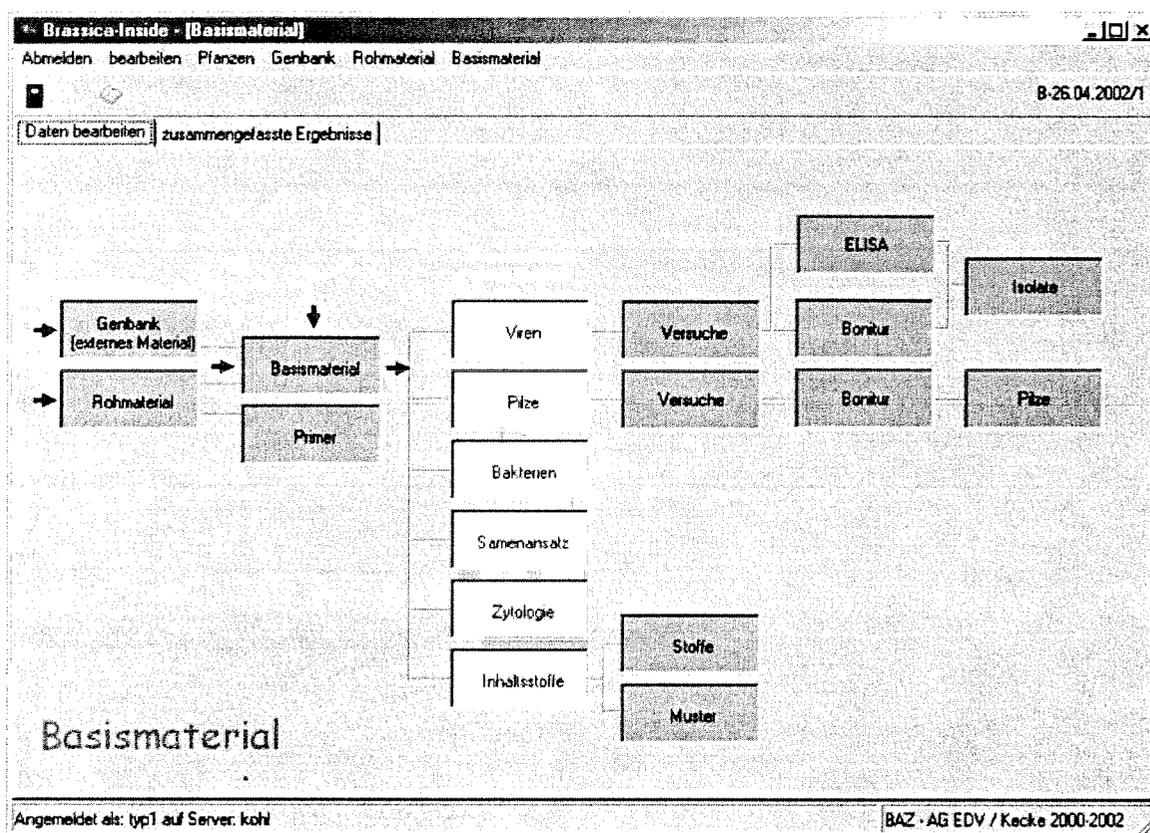


Abb. 4: Startbildschirm der Gruppe Basismaterial

Die Gruppe Basismaterial fungiert als „Verwalter der Pflanzen“ im Projekt und nimmt damit eine zentrale Stellung ein. Zu ihren Aufgaben gehören:

- Genbankmaterial erzeugen
- Rohmaterial erzeugen (eingeschränkt, da Aufgabe der Gruppe 2)
- Genbankmaterial bearbeiten
- Rohmaterial bearbeiten
- Basismaterial erzeugen
- Quereinstieg – Eingabe Basismaterial
- Basismaterial löschen (wenn noch nicht weitergegeben)
- Pflanzen (Basismaterial) weitergeben

Beispielhaft seien in den Abb. 5 und 6 die Bearbeitungsdialoge für die Daten der Genbank- und Basismaterialpflanzen gezeigt.

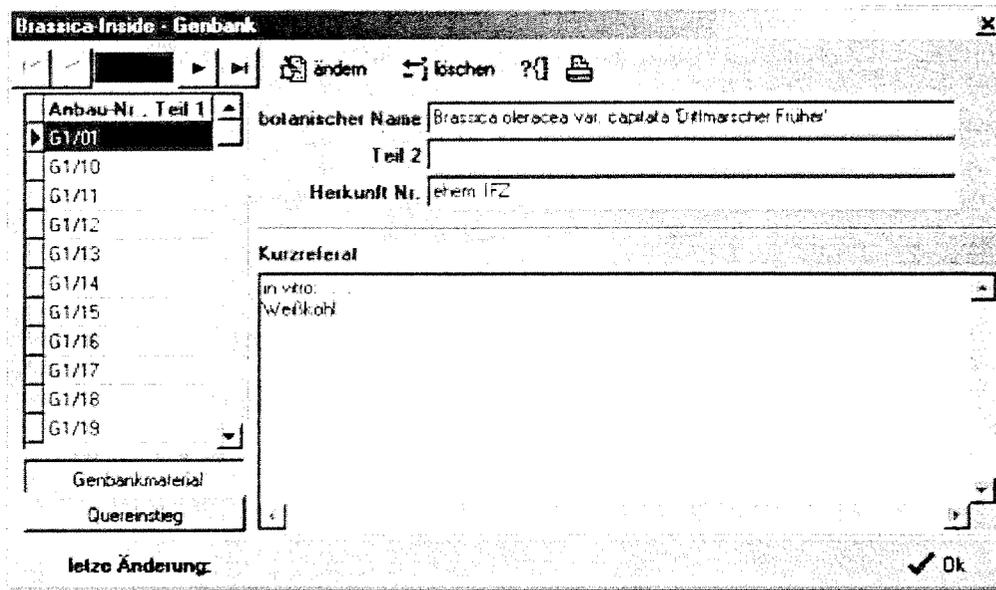


Abb. 5: Bearbeitungsdialog für externes Material (z. B. aus Genbank)

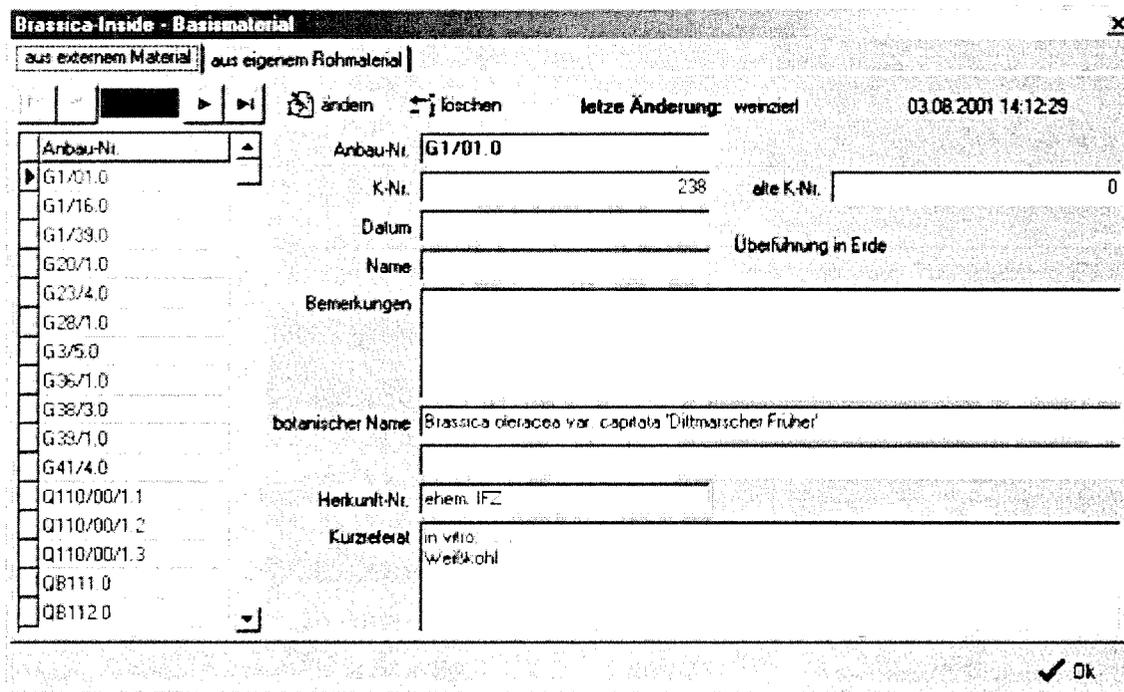


Abb. 6: Bearbeitungsdialog Basismaterial

### 2.2.2 Gruppe 2 – Rohmaterial

Die Nutzer der Gruppe 2 speisen so genanntes Rohmaterial in das System ein. Es werden entweder aus bereits im System vorhandenem Basismaterial (Einzelpflanzen) oder aus externen Pflanzen neue Rohmaterialpflanzen erzeugt. Dies geschieht mit unterschiedlichen Methoden, wie z. B. der Protoplastenfusion. Das externe Material stammt in erster Linie aus Genbanksammlungen. Nach erfolgter Prüfung des neuen Rohmaterials durch die Gruppe 3 werden die Pflanzen geklont und damit aus dem Rohmaterial neue Basismaterialpflanzen erzeugt.

Zu den Aufgaben der Gruppe 2 gehören somit:

- Genbankmaterial erzeugen
- Rohmaterial erzeugen
- Genbankmaterial bearbeiten
- Rohmaterial bearbeiten
- Basismaterial erzeugen

In Abb. 7 und 8 werden die Dialoge für die Funktionen „Erzeugen von Rohmaterial“ und „Überführung in Basismaterial“ gezeigt.

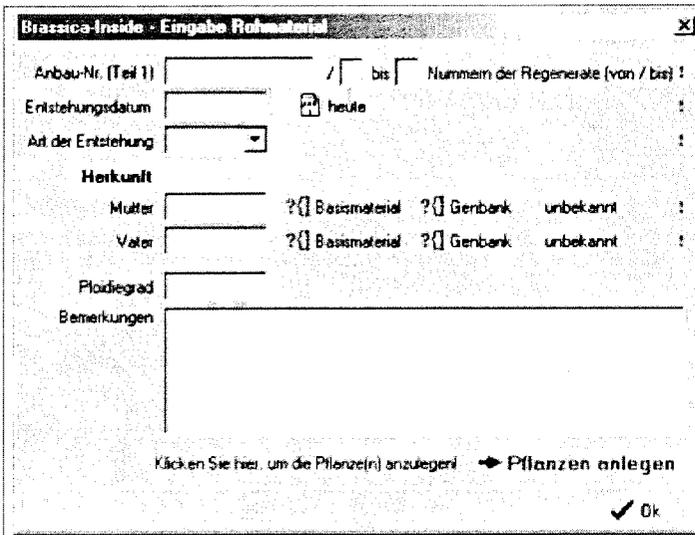


Abb. 7: Erzeugen von Rohmaterial

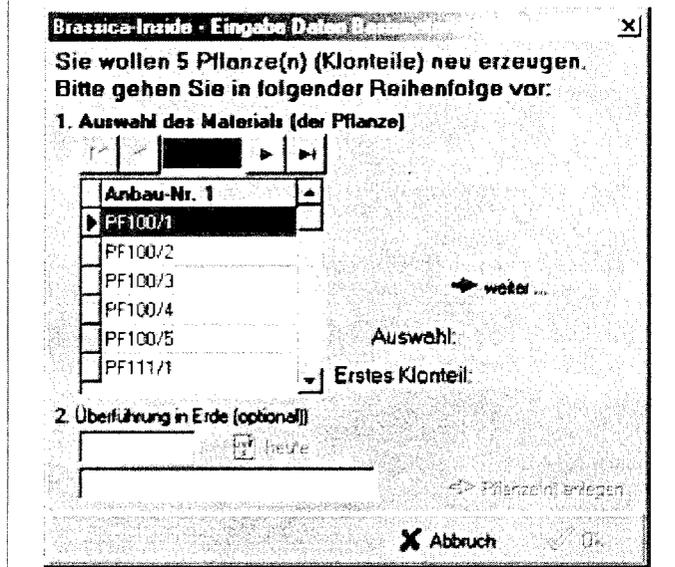
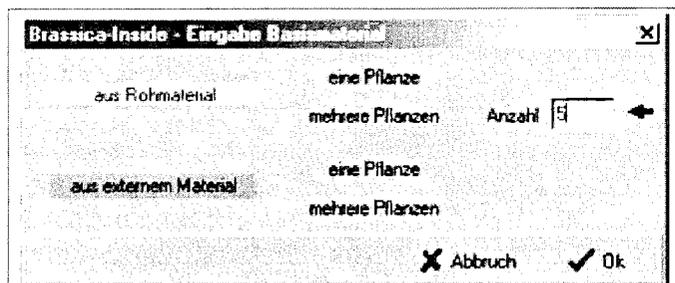


Abb. 8: Überführung in Basismaterial

Das „Erzeugen von Rohmaterial“ beschreibt die Fusion einer neuen Pflanze mit allen erforderlichen Daten, wie z. B. den Angaben zu den Eltern. Damit ist es an anderer Stelle im Programm möglich, zu allen Pflanzen ihre Stammbäume in beliebiger Tiefe rückwärts zu ermitteln und diese anzuzeigen und auszuwerten.

Die „Überführung in Basismaterial“ ist ein wesentlicher Schritt im System, denn hier wird die Verklonung von erfolgreich getestetem Rohmaterial zur Vervielfältigung der Pflanzen beschrieben. Bei der Überführung von Rohmaterialpflanzen in Basismaterial bietet ein erster Dialog die Möglichkeit der Auswahl zwischen Rohmaterial- und externen Pflanzen z. B. aus einer Genbanksammlung. Weiterhin wird hier die Zahl der Klone festgelegt.

Erst im folgenden Dialog (Abb. 8) werden – abhängig von den Eingaben im ersten Dialog – die Daten für die Überführung eingegeben. Wenn die erforderlichen Felder ausgefüllt sind, wird die Schaltfläche für das Anlegen jeder einzelnen Pflanze aktiviert. Das Anlegen von Basismaterial wird, wie auch andere Programmaktionen in einem separaten Fenster zur Kontrolle protokolliert. Die Protokolle können gespeichert oder als Bericht ausgedruckt werden.

Durch das Anklicken der Schaltfläche „Pflanze(n) anlegen“ werden im Hintergrund alle erforderlichen Datenbankaktionen ausgeführt.

Streng genommen müsste jede Pflanze, die Eingang in das System finden soll, aus externem Material oder bereits früher erzeugtem Basismaterial über die gezeigten Programmfunktionen eingegeben werden. Da nun aber das Projekt parallel zu laufenden Arbeiten entwickelt wird, und vor allem viele der Programmfunktionen für bereits vorhandenes Pflanzenmaterial genutzt werden sollen, war es erforderlich, dafür einen „Seiteneingang“ vorzusehen.

Dieser ermöglicht es den Nutzern der unterschiedlichen Gruppen, bereits vorhandenes Basismaterial per „Quereinstieg“ einzugeben. Erkennbar sind diese Pflanzen an der Anbaunummer. Diese beginnt hier mit dem Buchstaben „Q“ und einem weiteren als Kennzeichen für die jeweilige Gruppe. Die Anbaunummer QV01/00/2.6 deutet also beispielsweise darauf hin, dass diese Pflanze von einem Nutzer der Gruppe „Viren“ per Quereinstieg eingegeben wurde.

In der Datenbank werden diese Pflanzen wie solche behandelt, die aus externem Material eingegeben wurden. Ein entsprechender Datensatz wird in der Tabelle „genbank“ automatisch angelegt. Als negativer Effekt dieses Aufbruchs des ansonsten durchgängigen und lückenlosen Systems der Dokumentation des Lebensweges jeder einzelnen Pflanze muss jedoch in Kauf genommen werden, dass hier die Rückverfolgung von Stammbäumen immer aufhört, da über die Vorfahren von Quereinstiegspflanzen nichts dokumentiert werden kann.

### **2.2.3 Gruppe 3 – molekulare Marker**

Nachdem die Rohmaterialpflanzen erzeugt wurden, werden sie von Nutzern der Gruppe 3 z. B. mittels RAPD-Analyse geprüft. Dabei wird festgestellt, ob es sich um echte Fusionate handelt und die Pflanzen weiterverwendet und vermehrt werden können. Abb. 9 zeigt den Dialog zur Eingabe und Bearbeitung der Daten zu den RAPD-Analysen.

Die Ergebnisse der Analyse werden in die Datenbank eingetragen, so dass die Nutzer der Gruppe 2 erkennen können, welche Fusionen gelungen sind.

Wie auch in den folgenden Gruppen, so können die Mitarbeiter aus der Gruppe 3 unabhängig von den Pflanzendaten das System nutzen, um ihre eigenen Ergebnisse effektiv zu speichern und zu dokumentieren. Mit Hilfe spezieller Analyse- und Auswertungsprogramme können später weiterführende gruppenspezifische Datenauswertungen vorgenommen werden.

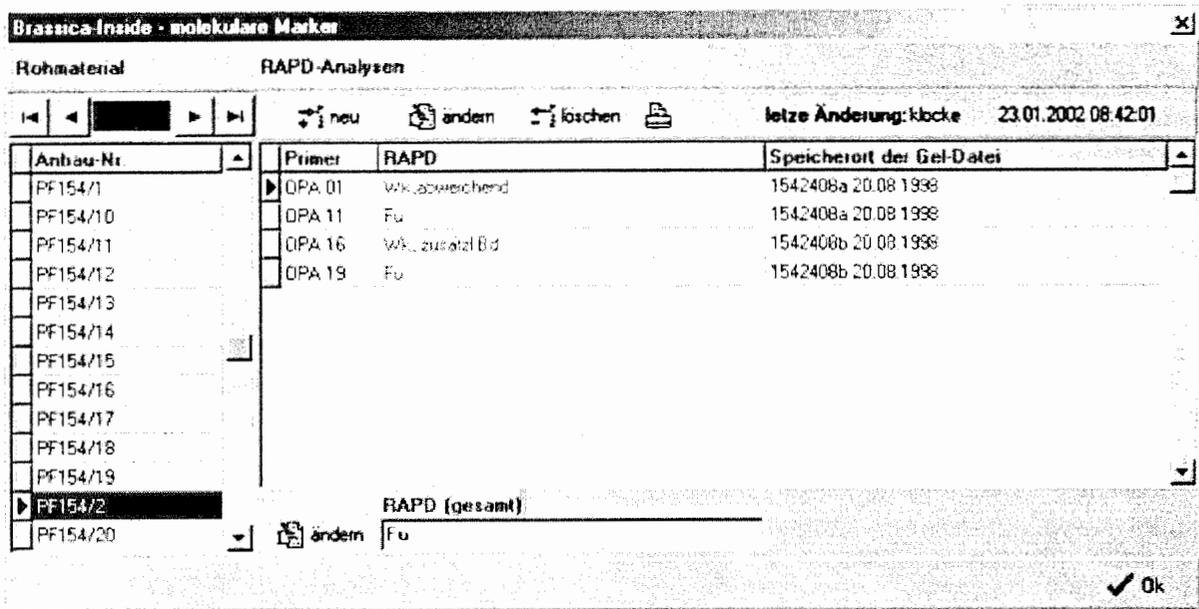


Abb. 9: Dialogfenster RAPD-Daten

### 2.2.4 Gruppe 4 – Viren

Nachdem die Pflanzen in vitro vermehrt und in Erde überführt worden sind, werden sie von der Gruppe 1 an die weiteren Arbeitsgruppen aufgeteilt. Die Gruppe 4 führt mittels ELISA-Tests und Bonituren Virusresistenzprüfungen durch.

Bei dieser und der Gruppe 5 sieht der Ablauf vor, dass zuerst Versuche definiert werden (Abb. 10). Diesen werden nachfolgend die Pflanzen zugeordnet. Ein Versuch definiert sich in der Gruppe 4 als die Menge aller Pflanzen mit gleichem Inokulationsdatum. Dabei können die Pflanzen durchaus mit unterschiedlichen Isolaten inokuliert sein.

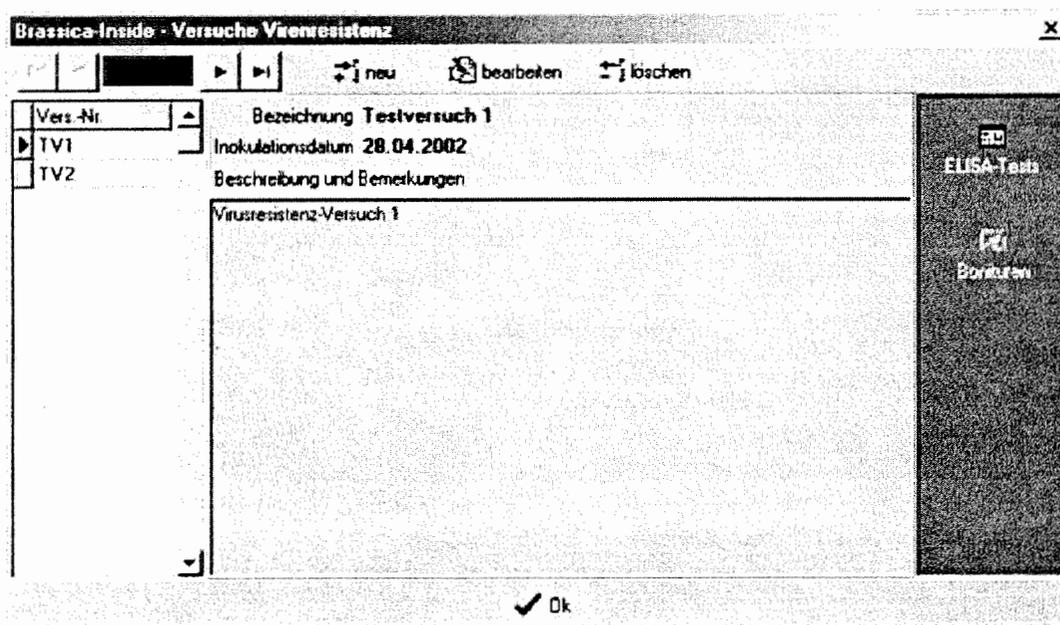


Abb. 10: Bearbeitungsdialog Versuche Virusresistenzprüfungen

Um den Aufwand der Dateneingabe zu minimieren, werden die ELISA-Daten als Datenfile direkt eingelesen. In Abb. 11 wird das Programmfenster mit den Daten eines kompletten ELISA-Tests dargestellt.

Zusätzlich zu den bereits in Zweiergruppen zusammengefassten ELISA-Daten werden bereits die Mittelwerte der Zweiergruppen, ein so genannter Schwellenwert, Kontrollmittelwerte u.a. errechnet und angegeben. Mit diesen Daten werden dem Versuchsansteller bereits aufbereitetes Zahlenmaterial und Resistenzaussagen nach verschiedenen Methoden in die Hand gegeben, deren manuelle Berechnung somit entfällt. Abb. 12 zeigt schematisch den Ablauf für einen Virusresistenzversuch.

P...	Anbau-Nr.	Pflanze	Mittel	Extinkt. 1	Extinkt. 2	R	R1	R2	Isolat
1		BLANK	0,017	-0,004	0,038	0	0	0	
3		PC	0,0345	0,51	1,159	1	1	1	
5		PC	1,1385	1,244	1,033	1	1	1	
7		PC	1,323	0,948	1,698	1	1	1	
9	PF100/4.1	PF100/4.1	0,642	0,588	0,696	1	1	1	11
11	PF100/4.2	PF100/4.2	0,1195	0,123	0,116	1	1	1	11
13	PF100/4.3	PF100/4.3	0,08	0,072	0,088	0	0	1	11
15	PF100/4.3	PF100/4.3	0,164	0,192	0,136	1	1	1	11
17		G1/D1.4	0,498	0,494	0,502	1	1	1	11
19		Lennox T1	0,0695	0,075	0,064	0	0	1	11
21		Lennox T1	0,3245	0,353	0,296	1	1	1	11
23		Lennox T1	0,204	0,208	0,2	1	1	1	11
25		Lennox T1	0,1295	0,127	0,132	1	1	1	11
27		Lennox T1	0,661	0,633	0,689	1	1	1	11
29		Lennox T1	0,154	0,193	0,115	1	1	1	11
31		Lennox T1	0,044	0,045	0,043	0	0	0	11
33		Lennox T1	0,562	0,491	0,633	1	1	1	11
35		Lennox T1	0,276	0,268	0,284	1	1	1	11
37		Lennox T1	0,0505	0,052	0,049	0	0	0	11

Abb. 11: Ergebnisdaten ELISA-Test

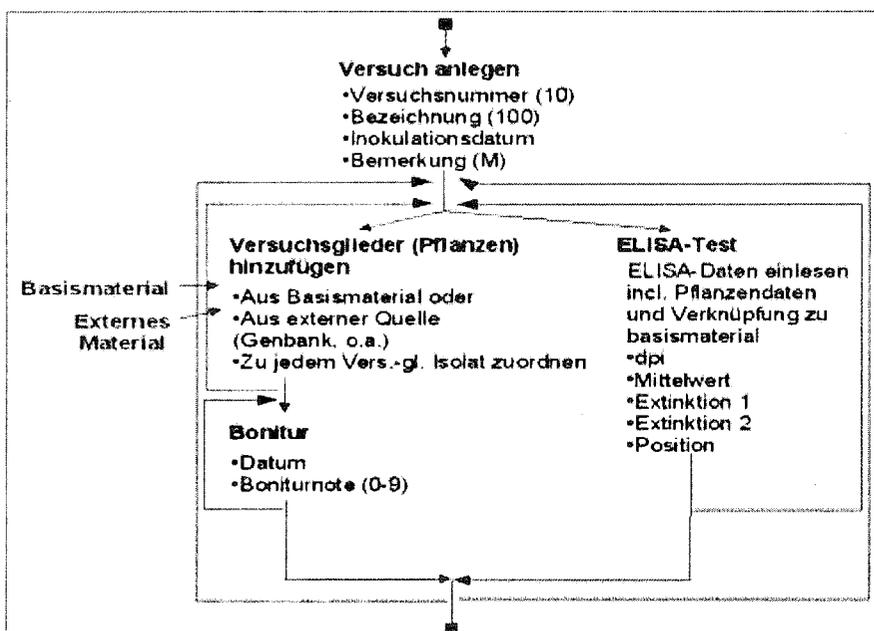


Abb. 12: Ablaufschema Virusresistenzversuch

### 2.2.5 Gruppe 5 – Pilze

Die Programmfunktionen für die Gruppe 5 ähneln in ihrer Handhabung denen der Gruppe 4. Auch hier

werden Versuche angelegt, denen Pflanzen zugeordnet werden können. Entsprechend dem Versuchsgeschehen können Pflanzen mehrfach bonitiert werden. Das ist z. B. erforderlich, wenn der zeitliche Verlauf der Krankheit beobachtet werden soll.

In den Abbildungen 13 und 14 werden die Dialogfenster zur Bearbeitung der Bonituren bzw. zur Eingabe der Boniturnoten gezeigt. Die Spalte KI in der Tabelle in Abb. 13 beinhaltet den so genannten Krankheitsindex, der auf der Basis der vergebenen Boniturnoten automatisch berechnet wird. Die Berechnungsvorschrift entspricht den Wünschen der Gruppe Pilzresistenz. Auch hier trägt das System durch automatische Aufbereitung der Versuchsergebnisse zur schnellen, fehlerfreien und effektiven Versuchsauswertung und -dokumentation bei.

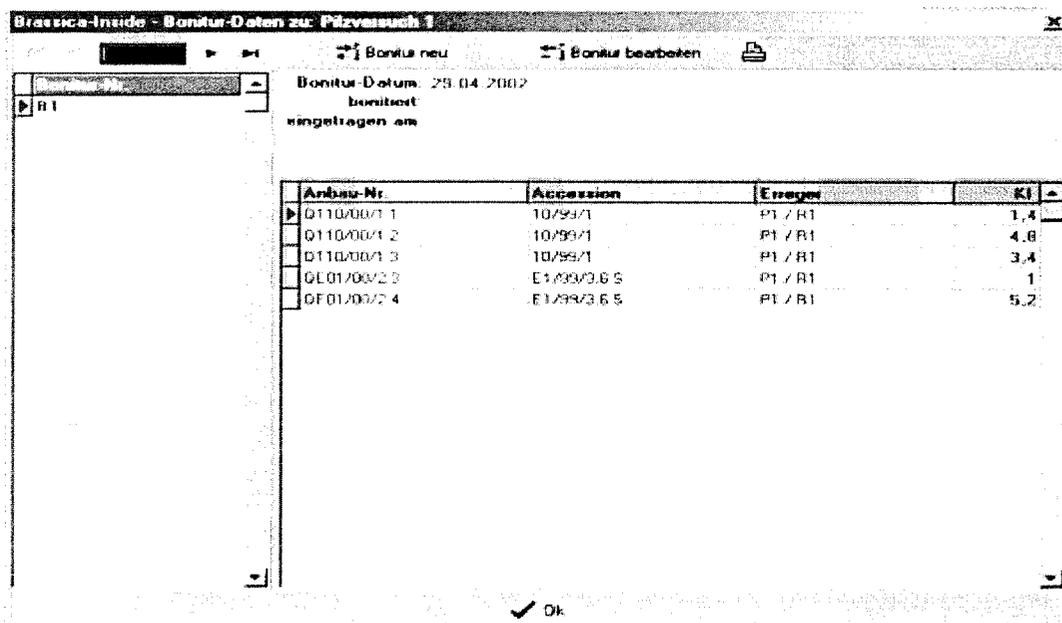


Abb. 13: Dialogfenster Bonituren Pilzresistenz

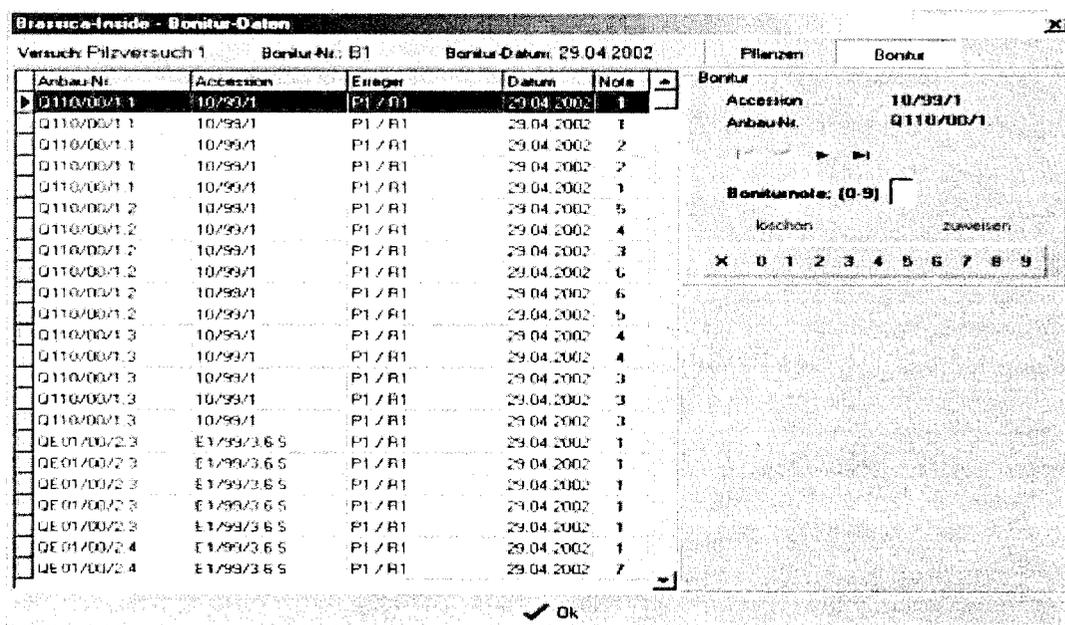


Abb. 14: Dialogfenster Bonitur

## 2.2.6 Auswertungen

Im Rahmen dieses hier erläuterten Eingabe- und Bearbeitungsprogrammes für die Daten ist lediglich eine vereinfachte übersichtliche und gruppenübergreifende Auswertung der Daten vorgesehen. Abb. 15 zeigt den interaktiven Dialog, in dem man alle gespeicherten Informationen für jede einzelne Pflanze abrufen kann. Darüber hinaus ermöglicht es das Programm, für jede Pflanze den Stammbaum anzuzeigen sowie den Weg der Pflanze durch die einzelnen Arbeitsgruppen zu dokumentieren.

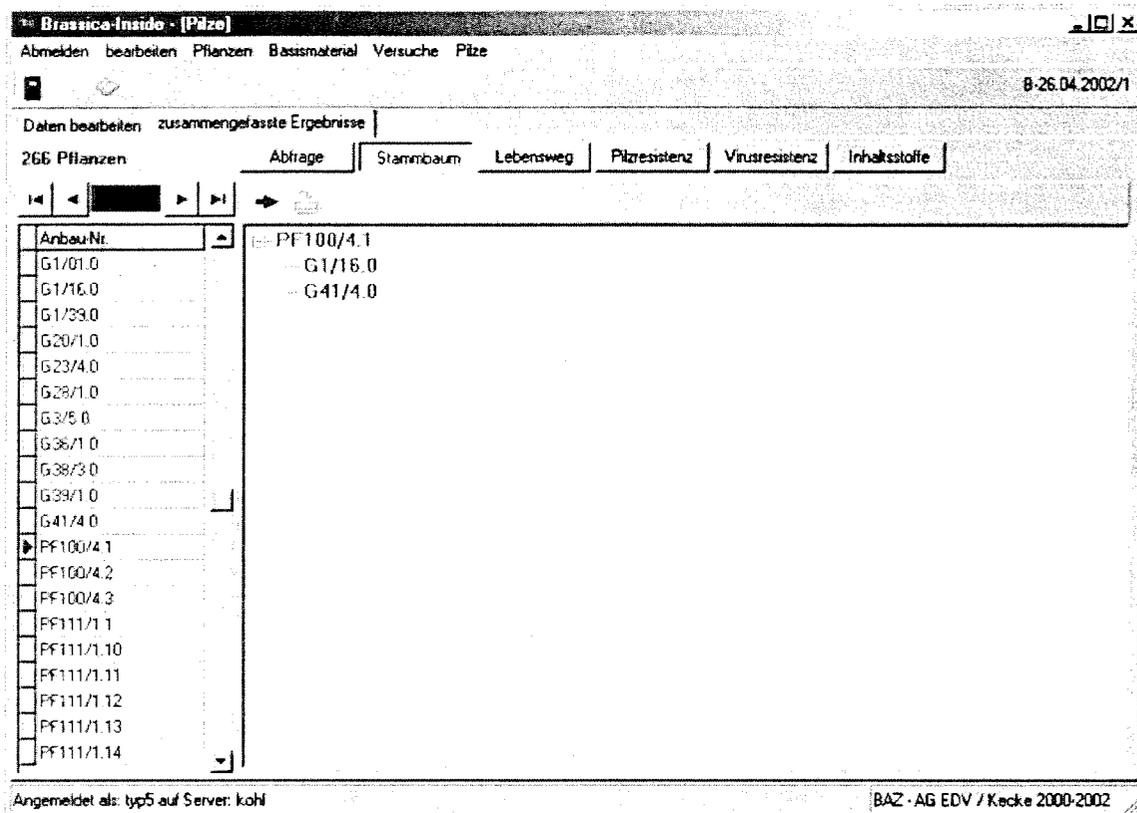


Abb. 15: Dialogfenster Auswertung

## 3 Aktueller Stand und weitere Entwicklung

Das Projekt befindet sich für die einzelnen Gruppen in unterschiedlichen Bearbeitungsstufen. Während die Gruppen 1 bis 3 bereits seit einiger Zeit laufende Daten eingeben, testen die Gruppen 4 und 5 zur Zeit die Handhabung des Systems. Zu diesem Zweck wurde ein separater Testdatenbankserver installiert. Die Programmfunktionen für die Gruppe 6 befinden sich noch in der Entwicklung.

Wie bei komplexen und für die beteiligten Fachwissenschaftler neuartigen Datenmanagementsystemen nicht anders zu erwarten, flossen während der bisherigen Entwicklungsarbeit, die immer wieder durch Diskussion von Teilproblemen begleitet wurde, eine Menge neuer Ideen und Anregungen in das Projekt ein. Es entstanden Ideen für neue, weitergehende Projekte, da sie den zeitlichen Rahmen des laufenden sprengen und aus der konkreten Aufgabenstellung zu weit herausführen würden.

In der folgenden Liste werden die wichtigsten weiteren Vorhaben genannt:

- Typ Inhaltsstoffe: Realisierung des Moduls
- Modellierung und Realisierung einer zentralen Schaderreger-Datenbank (neues Projekt)
- Einbinden der Schaderreger-Datenbank in die Module Viren und Pilze (Ersatz der bisherigen Viren- und Pilze-Tabellen)
- Erweiterung der Datenbank für weitere Kulturarten (neues Projekt)

- Dezentralisierung, um auch an weiteren BAZ-Standorten einzelne Module nutzen zu können (Bakterien in Aschersleben)
- Ausbau der Abfrage- und Auswertungsmöglichkeiten mit Datenexport (z. B. SAS-Input)
- Beschreibung und Ablage von Auswertungsmethoden (z. B. SAS-Skripte, die zusammen mit den extrahierten Daten übergeben werden)

Ursprünglich geplante, aber wegen der Eigendynamik des Projektes aus Kapazitätsgründen bisher nicht umgesetzte Vorhaben der Untersuchung von Methoden der Wissensverarbeitung zur Erschließung bisher nicht erkennbarer Zusammenhänge werden Anschlussprojekten vorbehalten bleiben. Biometrische Methoden, Neuronale Netze und Data Mining wären erfolgversprechende Ansatzpunkte.

Entsprechend der sich abzeichnenden Zielstellung, in der BAZ ein zentrales Datenmanagement für Forschungsdaten aufzubauen, werden sich die weiteren Arbeiten an diesem Projekt an bereits erkennbaren diesbezüglichen Randbedingungen und Schnittstellendefinitionen orientieren. Damit wird die Integration in künftige Systeme vorbereitet und sichergestellt.

## Literatur

DUBOIS, P. (2000): MySQL Entwicklung, Implementierung und Referenz  
Markt+Technik

KEGLER, H.; FRIEDT, W. (1993): Resistenz von Kulturpflanzen gegen pflanzenpathogene Viren  
Gustav Fischer Verlag, 175-176

MATTHIESEN, G., UNTERSTEIN, M. (1997): Relationale Datenbanken und SQL. Konzepte der Entwicklung  
und Anwendung  
Addison-Wesley

STAAB, S. (2002): Wissensmanagement mit Ontologien und Metadaten  
Informatik-Spektrum **25** (3), 194-209



**Anwesenheitsliste**  
**Workshop Datenmanagement 30. April 2002**

Brüggemann, Jörg, Dr.	Bundesanstalt für Getreide-, Kartoffel- und Fettforschung	j.brueggemann@bagkf.de
Bülow, Lorenz, Dr.	Technische Universität Braunschweig	lbuelow@tu-bs.de
Ehrhardt, Siegfried A	Bundesanstalt für Fleischforschung	z-ehhardt@baff-kulmbach.de
Germeier, Christoph, Dr.	Bundesanstalt für Züchtungsforschung an Kulturpflanzen	c.germeier@bafz.de
Hennig, Petra	Bundesforschungsanstalt für Forst- und Holzwirtschaft	hennig@holz.uni-hamburg.de
Jacobs, Helen	Institut für Agrartechnik Bornim e.V.	hjacob@atb-potsdam.de
Kecke, Steffen	Bundesanstalt für Züchtungsforschung an Kulturpflanzen	s.kecke@bafz.de
Kellermann, Hans-Jürgen, Dipl.-Phys.	Bundesforschungsanstalt für Fischerei	kellermann.ifo@bfa-fisch.de
Klöß, Detlef	Bundesforschungsanstalt für Viruskrankheiten der Tiere	d.kloess@wus.bfav.de
Kroschewski, Klaus, Dr.	Bundesforschungsanstalt für Viruskrankheiten der Tiere	klaus.kroschewski@wus.bfav.de
Langer, Michael	Bundesforschungsanstalt für Landwirtschaft	michael.langer@fal.de
Liesebach, Mirko, Dr.	Bundesforschungsanstalt für Forst- und Holzwirtschaft	liesebach@holz.uni-hamburg.de
Moll, Eckard, Dr.	Biologische Bundesanstalt für Land- und Forstwirtschaft	e.moll@bba.de
Roggendorf, Wolfgang, Dipl.Ing.agr.	Bundesforschungsanstalt für Landwirtschaft	wolfgang.roggendorf@fal.de
Schröder, Ronald	Bundesforschungsanstalt für Viruskrankheiten der Tiere	r.schroeder@wus.bfav.de
Schwarz, Jürgen, Dipl. Ing. sc. agr.	Institut für Agrartechnik Bornim e.V.	jschwarz@atb-potsdam.de
Sellmann, Jörg	Biologische Bundesanstalt für Land- und Forstwirtschaft	j.sellmann@bba.de
Staubach, Christoph, Dr.	Bundesforschungsanstalt für Viruskrankheiten der Tiere	staubach@wus.bfav.de
Stauber, Thomas	Bundesforschungsanstalt für Forst- und Holzwirtschaft	stauber@holz.uni-hamburg.de
Worseck, Stephan, Dr.	Biologische Bundesanstalt für Land- und Forstwirtschaft	s.worseck@bba.de
Ziller, Mario, Dr.	Bundesforschungsanstalt für Viruskrankheiten der Tiere	mario.ziller@wus.bfav.de

**Berichte** aus der Biologischen Bundesanstalt für Land- und Forstwirtschaft  
erscheinen seit 1995 in zwangloser Folge.

- Heft 100, 2002: EU-Beurteilungsbericht Flupyrsulfuron-methyl. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 29.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 101, 2002: EU-Beurteilungsbericht *Paecilomyces fumosoroseus*. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 30.  
Bearbeitet von Dr. Martina Erdtmann-Vourliotis, Dr. Axel Wilkening und Susanne Schaper, getr. Zählung.
- Heft 102, 2002: EU-Beurteilungsbericht Isoproturon. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 31.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 103, 2002: Zuständigkeiten bei der Prüfung und Zulassung von Pflanzenschutzmitteln und bei der EU-Wirkstoffprüfung.  
Stand: Februar 2002. Bearbeitet von Edelgard Adam, 58 S.
- Heft 104, 2002: Pflanzenschutz im ökologischen Landbau – Probleme und Lösungsansätze.  
Sechstes Fachgespräch am 26. Juni 2001 in Braunschweig. Abwehr von Wühlmausschäden im ökologischen Landbau. Bearbeitet von Dr. Hans-Joachim Pelz, 109 S.
- Heft 105, 2002: EU-Beurteilungsbericht Acibenzolar-S-methyl. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 32.  
Bearbeitet von Herbert Köpp und Susanne Schaper, getr. Zählung.
- Heft 106, 2002: EU-Beurteilungsbericht Eisen(III)phosphat. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 33.  
Bearbeitet von Dr. Martina Erdtmann-Vourliotis, Dr. Axel Wilkening und Susanne Schaper, getr. Zählung.
- Heft 107, 2002: EU-Beurteilungsbericht Ethofumesat. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 34.  
Bearbeitet von Dr. Martina Erdtmann-Vourliotis und Susanne Schaper, getr. Zählung.
- Heft 108, 2002: EU-Beurteilungsbericht Cyclanilide. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 35.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 109, 2002: Alternativen zum Einsatz von kupferhaltigen Präparaten im Apfelanbau. Ergebnisse einer Literaturrecherche.  
Bearbeitet von Dr. Beate Golba in Zusammenarbeit mit dem Institut für Pflanzenschutz im Obstbau der Biologischen Bundesanstalt für Land- und Forstwirtschaft, 67 S.
- Heft 110, 2002: Bewertungskonzept zum Nahtransport von Pflanzenschutzmitteln infolge Exposition über den Luftpfad (Abtrieb, Verflüchtigung und Deposition). Dr. Reinhard Winkler, Dr. Rainer Binner, Dr. Dietmar Gottschild, Dr. Wolfgang Koch und Dr. Johannes Siebers, 19 S.
- Heft 111, 2002: EU-Beurteilungsbericht Iprovalicarb. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 36.  
Bearbeitet von Herbert Köpp und Susanne Schaper, getr. Zählung.
- Heft 112, 2002: EU-Beurteilungsbericht Prosulfuron. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 37.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 113, 2002: EU-Beurteilungsbericht Pymetrozin. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 38.  
Bearbeitet von Dr. Martina Erdtmann-Vourliotis, Dr. Axel Wilkening und Susanne Schaper, getr. Zählung.
- Heft 114, 2002: EU-Beurteilungsbericht Pyraflufen-ethyl. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 39.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 115, 2002: EU-Beurteilungsbericht Sulfosulfuron. Rechtliche Regelungen der Europäischen Union zu Pflanzenschutzmitteln und deren Wirkstoffen. Band D 40.  
Bearbeitet von Dr. Henning Bruno und Susanne Schaper, getr. Zählung.
- Heft 116, 2002: Liste der zugelassenen Pflanzenschutzmittel (Stand: 1. Juli 2002).  
Bearbeitet von Andreas Spinti, 78 S.
- Heft 117, 2002: Fachgespräch „Anwendungsbestimmungen für Pflanzenschutzmittel zum Schutz von aquatischen und terrestrischen Biozönosen (Flora und Fauna) in der Praxis – ein Erfahrungsaustausch“.  
Bearbeitet von Dr. Rolf Forster, 68 S.
- Heft 118, 2003: Pflanzenschutz im ökologischen Landbau – Probleme und Lösungsansätze.  
Siebtes Fachgespräch am 6. Juni 2002 in Berlin-Dahlem. Alternativen zur Anwendung von Kupfer als Pflanzenschutzmittel. Forschungsstand und neue Lösungsansätze.  
Bearbeitet von PD Dr. habil. Stefan Kühne und Britta Friedrich, 69 S.