Gregor Hagedorn

Making DELTA Accessible: Databasing Descriptive Information

Abstract

Hagedorn,G.: Making DELTA Accessible: Databasing Descriptive Information. – Bocconea 13: 261-280. 2001. – ISSN 1120-4060.

The general concept of designing biological databases from modular applications is presented and exemplified by DeltaAccess, a database subsystem for descriptive information in taxonomy. DeltaAccess makes DELTA, the "Description Language for Taxonomy", accessible to the world of relational databases and to modern graphical user interfaces. The advantages of DELTA compatible database applications are outlined in comparison with conventional DELTA programs. Using a database for managing descriptive information opens exciting new perspectives for local as well as international collaborative projects. Specific database features important for such projects (like subproject views, summarizing data, and data replication) are discussed in more detail. The subject is relevant to anyone planning large-scale projects involving taxonomy, interactive identification, or checklist data in botany or zoology.

Introduction

The present article discusses concepts and compares tools to manage descriptive data in biology. The concept of descriptive data is defined and the important concept of information systems built from modular database subsystems is introduced. After a general overview of the DELTA data format and the advantages of using a DELTA compatible software package, the additional advantages of using a DELTA compatible database subsystem instead of a conventional program are outlined. The database application DeltaAccess (Hagedorn 1997-2001) is used to illustrate these advantages.

The article focuses on certain features of DeltaAccess that can only be provided because DeltaAccess relies on an underlying database management system. Features like multi-user operation, data security, database replication, dynamic views, and linked projects allow scientists to collaborate and share information in ways which are unavailable in conventional DELTA compatible software. Some of these features are not only available in DeltaAccess, but in other DELTA compatible database applications as well. The article therefore provides important background information to assist the reader in deciding which features are required and which software product best meets their needs for a given project.

What are descriptive data?

In biology, descriptive data are records of features or properties of organisms. They include morphological and anatomical as well as physiological and molecular data (e.g. DNA sequences). Descriptive data can be the result of experiments or observations. In both cases, the raw data are usually further analyzed and interpreted, using the inductive and the comparative method, respectively. Theoretically, these analyses form a separate layer of information, the synthesis. In practice, part of the synthesis is incorporated into the recorded descriptive data, since the concepts and terminology used for description are the result of synthetical processes. For example, in a chromatographic analysis of secondary metabolites, the results are not usually cited as retention time and integration area; but instead, the raw data are analyzed and the name and amount of the substance are recorded.

In the case of a taxonomic monograph, multiple specimens or strains of a species are observed and the results summarized into synthetical descriptions for a whole species. These descriptions are usually presented in the form of so-called "natural language descriptions" ("leaves ovate, hairy, 10-30 cm long, ..."). Other parts of the work synthesize these data into taxonomical concepts, phylogenetic analyses, or identification keys.

Not all data in biology are descriptive. For example, the hierarchical systematic arrangement of taxa is a synthesis which is based on the analysis of descriptive data, but which is not identical with it. Management data, literature references, and the data on observations or collections of specimens should also be seen as separate kinds of data. Distribution data may either be viewed as descriptive data about range and occurrence, or may be viewed as a separate type of data.

The distinction between descriptive and non-descriptive data leads to the question of how biological database systems should be structured and to the concept of database subsystems.

Database subsystems

A common problem with current biological database systems is their monolithic design. While parts of existing applications may be excellent, other parts may be deficient or unsuitable for the needs of a specific project. One solution to this problem is to create an integrated biological database system built from modular applications. Such modules are called database subsystems in this article.

DeltaAccess is a descriptor database subsystem. Other important database subsystems concern literature references, nomenclature, taxonomy, specimen collections and observations, geography, and agents (persons, teams, organizations, etc.). By definition, a descriptor database subsystem should store no data pertaining to other subsystems except for linking information. For example, the scientific name of an organism can serve as a primary object identifier in a taxonomic database, allowing the retrieval of taxonomically relevant information about synonymy, combinations, or the type specimen. Fig. 1 gives an outline of the relation between the descriptor subsystem and examples of other subsystems. Links need not be limited to links between the descriptor database subsystem and other subsystems (see Fig. 2 for examples of links between subsystems).

Both the application and the information model of each subsystem should have well defined interfaces to other subsystems. Subsystems should be exchangeable modules. For

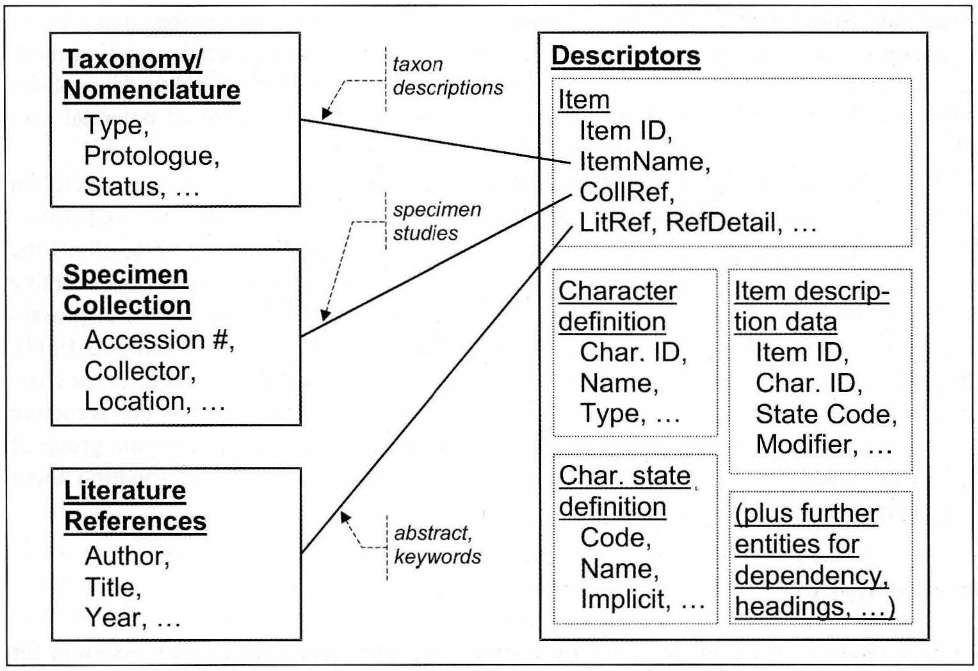

Fig. 1. Relationships between a descriptor database subsystem and other subsystems. Examples of attributes are listed for each subsystem. Each subsystem may consist of several entities; in the case of the descriptor subsystems the major entities are indicated as dotted boxes. Links can be read both ways. When seen from the taxonomy, specimen, and literature subsystem, the descriptors form a taxon description, specimen description, or keyword information, respectively. Seen from the descriptor subsystem, the links identify the sources of information and the object that is described.

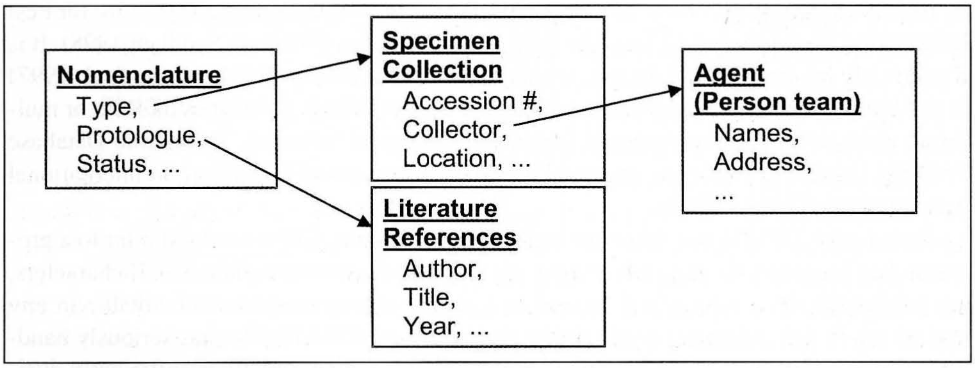

Fig. 2. Examples of relationships ("links") between subsystems. Note that although the author citation for a taxon can usually be assigned to a defined person or person team, this relationship can not be established for most literature references. No relationship is therefore assumed between authors of Literature references and Agents. Each author of a literature reference should be treated as a simple name; if a relationship definition to Agent is desired, it should be defined as an optional, secondary data element.

example, if the literature reference software does not fulfill the expectations and a better (perhaps commercial) product is found, it should be possible to replace the existing module without changing other parts of the database system. Another candidate for replacement could be the taxonomic subsystem, since the important concept of potential taxa (Berendsohn 1995) is not yet implemented in any available software.

While the attributes of specimen collections (herbaria, zoological collections, microbial culture collections) can be generalized even to the extent that living and conserved collections can be treated in a single information model (Berendsohn & al. 1996, 1999) the diverse taxonomic groups (e. g. mosses, viruses, and birds) have very few descriptive attributes in common. Even a relatively homogeneous group like the genera of grasses requires 540 characters to describe it (Watson & Dallwitz 1992, Watson & Dallwitz 1994). Since it is impossible to define a general list of descriptive attributes (= features, or characters) for all organisms, a descriptive database must be implemented as a metastructure capable of holding multiple schemes, each of which is appropriate for a certain group of items and questions. These schemes (or character definitions) contain information about the name and type of the attributes.

What is DELTA?

The "Description Language for Taxonomy" is a powerful data exchange format for descriptive data. Its origin goes back to work by Mike Dallwitz at Canberra University in 1973; it was first published in Dallwitz (1980). DELTA is probably the most widely used general-purpose format for descriptive data and it is used by several taxonomic software packages. The most well known are: "The Delta package" (containing Confor, Delfor, and Intkey; Dallwitz 1993, Dallwitz & al. 1995), Pankey and PANDORA (Pankhurst 1993, Pankhurst & Pullan 1996, Pankhurst 1998a, 1998b), TAXASOFT (Gouda 1999), and DeltaAccess (Hagedorn 1997-2001). DELTA is further supported with various restrictions by ALICE (White & al. 1993), BG-Base (BG-BASE Inc. 1997-99), LucID (Centre for Pest Information Technology and Transfer 1998), and CABIKey (White & Sandlant 1998). It is rivaled only by the NEXUS format, which in its current version 2 (Maddison & al. 1997) is still limited to analytical purposes and can not deal with basic data types like text or multistate characters. The International Union for Biological Sciences, Taxonomic Database Working Group (TDWG) has endorsed the basic directives of DELTA as an international data standard (TDWG 2000).

Technically, DELTA is a delimiter based free text format, which works similar to a programming language. Its main advantages are that data files contain plain ASCII characters, are comparatively compact, and remain to a certain degree readable and editable in any text editor. In fact, reliance on this ability to directly edit DELTA files has seriously handicapped the wider use of DELTA. Reading the DELTA coding directly is sufficiently difficult to prohibit most biologists from trying to use it. Two DOS-based DELTA editors are available (DEdit by R. Pankhurst and TAXASOFT by E. Gouda), which offer some guidance for editing DELTA projects. The first graphical, Windows-based editor appeared in 1997 with DeltaAccess. In 1998 the first beta version of an editor for Dallwitz's "Delta package" was released.

For the purpose of this article it is sufficient to regard DELTA as a data interchange format for descriptive data. More detailed information about the DELTA format can be found in Dallwitz & al. (1995) and chapter 5 of Pankhurst (1991).
In general, descriptive data can be classified according to their data type:

- *Categorical data* are used if only predefined categories ("character states") are to be scored for a given character. Categorical data are further classified as unordered (nominal scale, e. g. 'red'/'green'/'blue'/...) or ordered (ordinal scale, e. g. 'first'/'second'/'third'/...) data (see Fig. 3). Appropriate statistical methods exist only for unordered and linearly ordered categorical data, but non-linear relations between states (e. g. tree-like structures) contain valuable information and are supported by phylogenetic analysis programs like PAUP (Swofford 1990). Another important characteristic is whether a categorical character is exclusive (i.e. only a single state may be scored in each item) or multistate (i.e. several states may apply to a single item). Multistate characters are necessary to express the permanent presence of multiple states (e. g. a fungus may always have both septated and unseptated spores) or a variation between individuals (depending on the culture conditions, spores may have different shapes).
- *Numerical data* are commonly classified as integer (counts) or real numeric values (measurements). Original data (measurements, counts) and statistical reporting (mean, median, standard deviation, minimum, maximum, sample size, etc.) should be distinguished.
- *Date/time values and date/time ranges* can be seen as a special type of numerical data. In practice, these values are frequently only partially known ('20. May': year unknown, or 'May to June': day and year unknown), and special range features may be needed ('1888 or 1886' because handwriting is not readable). Date values are relevant in a descriptor database e.g. to record phenological data (including, e.g., the times of the year during which an adult insect is usually found flying).

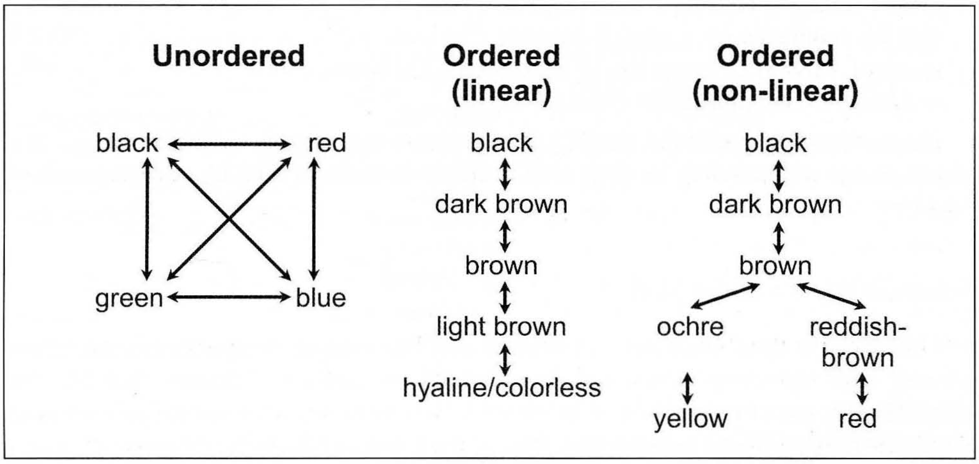

Fig. 3. Types of categorical data. The arrows between the character states indicate the possible transitions between states. While in an unordered categorical character each state can be directly transformed into each other state, this may require several steps in an ordered character. In the example, two steps are necessary to go from black to brown.

- *Text data* are indispensable, although textual descriptions can usually not be analyzed. A distinction should be made between long text (prose) and lists of short text values. To some extent short values can be analyzed and used in queries or for sorting purposes. Theoretically, these list characters could be replaced by a categorical character, but this is not very practical if thousands of states are possible. For example, the substrate of a microorganism may be any plant of the world as well as dead materials like leather, timber, or optical glass.
- *Graphics/audio/video data.* Illustrations (line drawings or photographs) are most important, but a general concept for all kind of media resources is necessary.
- Other object data with special properties, like DNA sequences, AFLP™ Patterns, etc.

The DELTA data standard can handle most of these data types, with various degrees of support. The shortcomings of the DELTA format are:

- Textual list values are not yet supported by DELTA (but this feature is planned for a new version of the DELTA format (Dallwitz & al. 1999).
- The support for numerical values is limited. No distinction is made between original data and statistical reporting. Sample size, standard deviation, and many other statistical parameters can not be stored other than as comments.
- Non-linear relations between states in ordinal categorical data are not supported. This is a disadvantage of DELTA compared with the NEXUS format, which supports this character type.
- Date and time values must be stored in text characters and are therefore not available for analysis and calculations.
- Although some support for graphical, audio, and video data has been added to Dallwitz's suite of DELTA programs, it is designed specifically to support a single program ('Intkey'). Hotspots, superimposable annotations, and buttons are encoded into DELTA comments by a special program ('Intimate') and are not part of the DELTA standard. Parts of an image can be associated with character states, but it is not possible to directly define images for character states.

Despite these shortcomings, the DELTA standard remains highly useful in practice. The shortcomings will certainly be dealt with in future versions of DELTA or a successor of DELTA.

What can DELTA do for you?

Collections of descriptive data, recorded in a DELTA compatible application can form a primary data repository, which can be evaluated for different purposes (see Fig. 4). Originally, the most important purposes were the construction of keys and the generation of taxon descriptions. These two elements form a major part of taxonomic revisions of, e.g., a genus or a family. Large monographs have been generated directly from DELTA files (see e. g. Watson & Dallwitz 1994). Other purposes, most notably interactive computer-aided identification, have later been added. Furthermore, it is possible to reformat the data from a DELTA file to several formats suitable for phylogenetic analysis, including distance matrices and the NEXUS format used by PAUP (Swofford 1990) and other programs. Without DELTA as a data exchange standard, it would be necessary to re-enter the data in the formats unique to each program, a process which is both laborious and error-prone.

Compiling data in a structured form initially requires more work than directly writing natural language descriptions, mostly because one must analyze and define the characters which will be used. However, this added effort can dramatically improve the quality of the work in general, because it forces the author to deal with inconsistencies in concepts and terminology, which might otherwise escape notice. Furthermore, certain characters are traditionally observed only in a small subset of a taxonomic group, where they are required for differentiation. Although DELTA can deal with missing data, the use of a DELTA compatible application encourages the researcher to record data more completely. This is very useful if the data set will be used for identification or data analysis (esp. phylogenetic analysis). Tabular or graphical reports can be generated to assess the completeness of the character scoring.

It is further possible to generate differential diagnoses of each taxon against all other taxa. Often, certain taxa appear to be insufficiently separated, although the author of the data set would intuitively consider the taxa well defined. Analyzing such discrepancies may help to improve the scientific quality of the information.

Another interesting aspect of using a character definition to generate natural language descriptions is that it is relatively easy to translate a data set into multiple languages. Once a character definition is translated, the major part of natural language descriptions and keys can be translated automatically. For each item only the text characters and notes must be individually translated. If most text characters contain language independent information like scientific names, places, or literature references, this work is further reduced. Several examples of translations into different languages can be found in Dallwitz (1993).

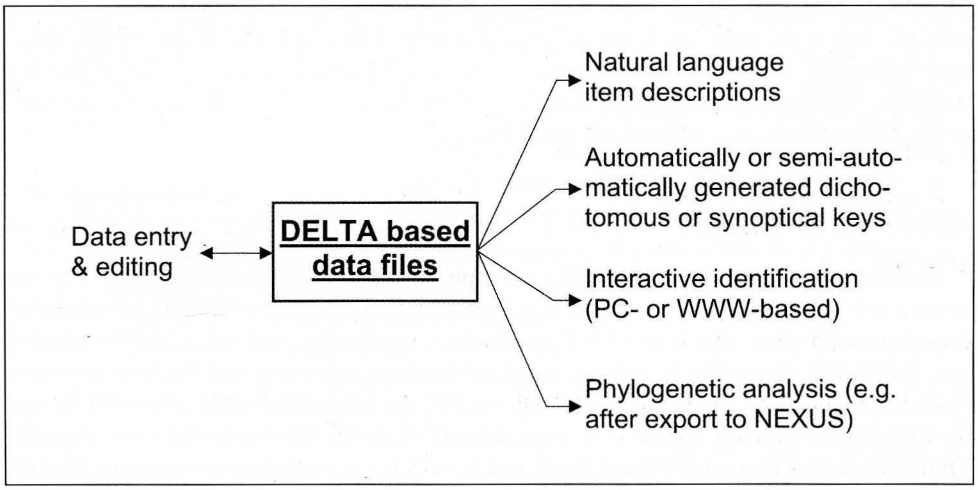
Data entry
& editing
DELTA based
data files
Natural language
item descriptions
Automatically or semi-automatically generated dichotomous or synoptical keys
Interactive identification
(PC- or WWW-based)
Phylogenetic analysis (e.g.
after export to NEXUS)

Fig. 4. Workflow schema of a conventional, text file based DELTA system. An editor imports and exports text files containing DELTA directives; other programs import this information to reformat it into reports or formats suitable for interactive identification programs or phylogenetic analysis. Interactive identification over the internet is possible, but requires each user to buy a licence for Intkey version 5.

What can a descriptor database do for you?

Conventional DELTA applications, such as Confor, Intkey, or TAXASOFT are dedicated programs which read the DELTA data exchange format, and either store it in memory alone or use a proprietary format to store the imported files on disk. In contrast to these file-based systems, DELTA compatible descriptor databases like PANDORA, DeltaAccess, or Alice import the DELTA data exchange format into a permanent database, accessible by standard methods (see Fig. 5).

If descriptive data are stored in a true database system, some additional features are available, which conventional DELTA programs cannot offer. These include multi-user data entry, editing and retrieval (several users can concurrently access the database, either in LAN or over the internet), the option to replicate multiple databases across the internet, and a full security model (see Fig. 5). Database features are especially important for large collaborative, international projects. Databases can provide the facilities to store and retrieve massive amounts of data and to record the contributions of multiple authors and editors to document the respective intellectual property rights. Using a database furthermore allows online analytical processing of the most recently edited data. The tight integration of editing with data retrieval and analysis tools can result in improved data quality and workflow.

It is not impossible to integrate these desirable features into a conventional DELTA program that uses dedicated data storage code, but it would be very costly to do so. It would mean re-implementing features from scratch that are worth hundreds of programming years. Using available database management systems enables the scientific community to profit from work undertaken primarily for business applications.

Several database applications are available which support DELTA to various degrees (PANDORA/Pankey, ALICE, and BG-Base). They differ in the amount of support for DELTA, and in the cost. They are integrated systems, not a specialized descriptive database subsystem as defined above. None of these applications is presently available for Windows. The descriptive database subsystem DeltaAccess, which is discussed more fully in the following chapter, is intended to fill this gap.

Introducing DeltaAccess

DeltaAccess (Hagedorn 1997-2001) attempts to make the software used to manage descriptive information more accessible to biologist, by providing a working environment consistent with other Windows 95/NT applications commonly used today, and by making descriptive data accessible to industry standard database, reporting, and analysis software. DeltaAccess is based on the relational data model, the most commonly employed design for information systems today. It is implemented using the PC-based database program Microsoft Access (versions 97 and 2000) and provides a complete environment to import, edit, analyze, and export DELTA based projects.

Most existing DELTA software packages operate as closed systems, interacting with other software only through import/export procedures. Export functions to standard analysis and reporting tools (e.g. spreadsheets, or statistical analysis software like SAS) are limited. It is impossible to dynamically link other data sources, like nomenclatorial, specimen,

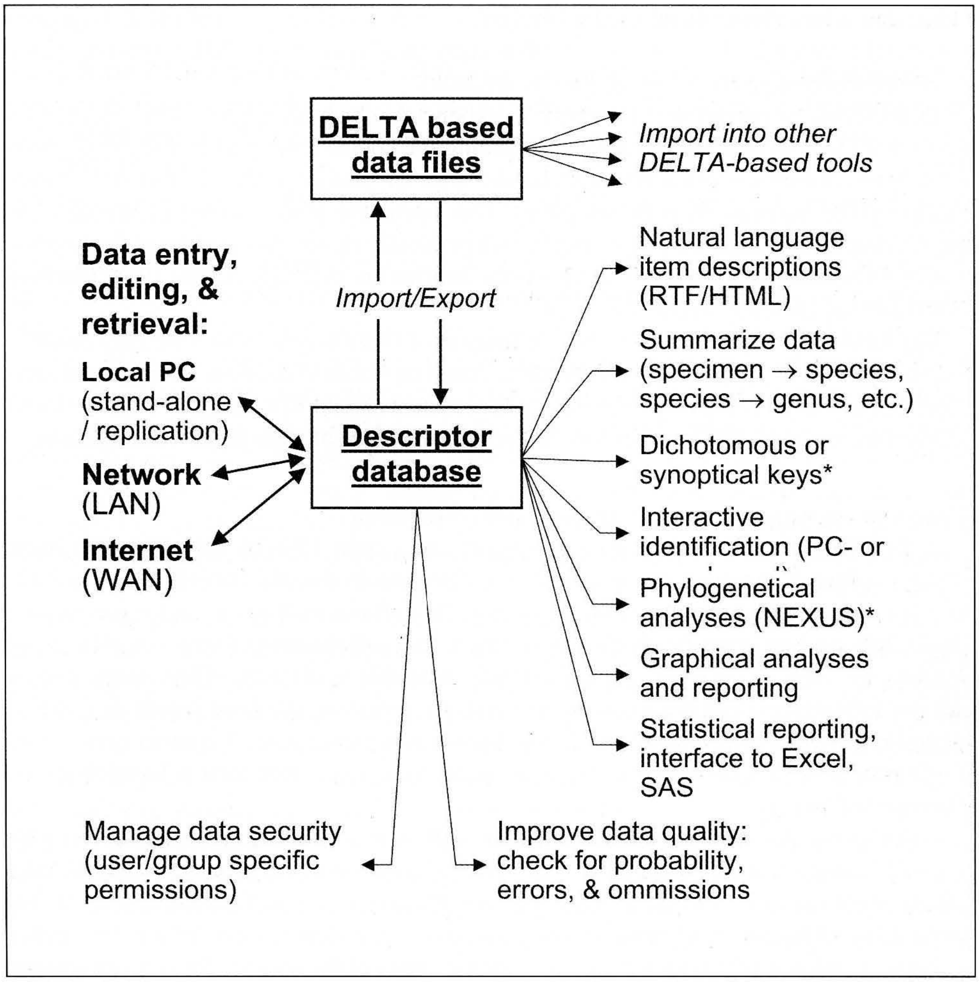
DELTA based data files
Import into other DELTA-based tools
Import/Export
Data entry, editing, & retrieval:
Local PC (stand-alone / replication)
Network (LAN)
Internet (WAN)
Descriptor database
Natural language item descriptions (RTF/HTML)
Summarize data (specimen → species, species → genus, etc.)
Dichotomous or synoptical keys*
Interactive identification (PC- or
Phylogenetical analyses (NEXUS)*
Graphical analyses and reporting
Statistical reporting, interface to Excel, SAS
Manage data security (user/group specific permissions)
Improve data quality: check for probability, errors, & ommissions

Fig. 5. Workflow schema of a system using a descriptor database like PANDORA or DeltaAccess. A '*' indicates features planned, but not yet implemented in DeltaAccess. DELTA text files can be imported and exported, but otherwise the data remain directly accessible in the database for online editing, analysis, and reporting. Important improvements include multi-user data entry, editing and retrieval (several users can concurrently access the database, either in LAN or over the internet), security management, and various analytical functions acting directly upon the database.

or literature reference database subsystems with the descriptive data. In contrast, DeltaAccess was designed to enable dynamic data interchange with standard PC databasing, reporting, and analysis applications as well as with network applications like internet web-servers and client-server databases. It can be used on a single PC, in a PC network, or in combination with any ODBC compliant database server (Oracle, SQL-Server, Sybase, etc.). DeltaAccess is explicitly designed to facilitate the sharing of information across different applications and improve collaboration among multiple researchers in a project.

DeltaAccess has already been used to analyze large data sets (e. g. Rambold & Hagedorn 1998) and to teach DELTA courses. The data entry facilities using HTML forms have been discussed in detail in Hagedorn & Rambold (2000).

DeltaAccess, is part of the "Diversity Workbench", which is currently being developed (see www.DiversityWorkbench.net). It including the source code and is distributed from www.bgbm.fu-berlin.de/projects/DeltaAccess free of charge under a General Public License (Free Software Foundation 2000). This opens the project to the community for improvements or additions. For example, independently developed interfaces for interactive identification and data retrieval across the internet are available (Cross 1998, Findling 1998) and are currently being improved.

The following chapters discuss specific features of DeltaAccess, which are only possible because it is a descriptor database subsystem, based on standard database management software. For any large collaborative project an assessment should first be made which of these features are required, and whether they are provided by the software under consideration.

Multi-user operation, local and international cooperation

A DELTA compatible database can import and export DELTA text files like other DELTA software, but once it is imported, the data remain directly accessible in the database for online editing, analysis, and reporting. This allows multiple users to work with a single data set concurrently, sharing information and collaborating in the compilation of the data set. New or updated data are instantly available to all users. Thus, many people can use interactive identification software, directly accessing the most recent data, while several researchers can continue to edit the data set at the same time. A conflict occurs only if two people try to edit the same character state in the same item (this is handled by the database software).

DeltaAccess can be used on a single PC as well as in a local area network (LAN) with up to 255 concurrent users. To serve wide area networks with many more users, the data can be migrated to a SQL database server and DeltaAccess can be used as a front-end application. Different user groups can be provided with different views of the data, either to focus attention on the relevant data or to protect parts of the data set for security reasons (see the section 'Projects, links, subsets, and views' below).

Starting with version 1.4, DeltaAccess is capable of generating HTML forms to edit descriptive data over the internet, allowing a client-server interaction between a database and a world-wide-web browser. The browser thus becomes the front end to the database. Although HTML forms are more limited than the DeltaAccess front end itself, this feature allows users outside the LAN or using non-Windows PCs to partake in the use and editing of the data.

Interoperability

It is not necessary that all tasks (identification, editing, reporting, and analysis) are performed by a single program. If a DELTA compatible application is based on a database, the descriptive data are available to at least all programs that can use the same database engine. Thus not only multiple users, but also multiple applications can access a data set concurrently. For instance, in the case of DeltaAccess, the underlying Microsoft JET database engine can be directly used by any application programmed in Microsoft Visual C or Visual Basic, and indirectly (through the ODBC or JDBC general data access standards) by a wide range of programs from many manufacturers. To develop a compatible application, it is not necessary to have the source code of the primary application; a basic understanding of the information model is sufficient.

Projects, links, subsets, and views

If multiple researchers co-operate in a project, it tends to contain a large number of characters (several hundred characters are common) and it may contain several thousand items. Such projects are often unwieldy, because it is difficult to find the right character and item. DELTA character definitions already provide a standard method to hide certain characters, by analyzing and declaring dependencies between characters. For example, if a plant has no leaves, leaf characters are irrelevant. Character dependencies are a very valuable feature, but they can hide irrelevant characters only once the parent characters ("leaves present") have been scored. No standard method to hide items while editing them is available.

However, an individual researcher will likely be primarily interested in a subset of the data set, e. g. in one or several genera out of a data set which comprises all species of a family. Restricting the view of this researcher to only the relevant items and characters should improve both the efficiency and the quality of this scientist's work.

The initial solution to achieve such a restricted view might be the creation of multiple small, independent projects (project A & B in Fig. 6). Doing so bears the danger of the character definitions becoming incompatible, even if several researchers initially agree upon a common list of characters. Subsequent analysis of character definitions to merge projects can be a very difficult task, because identical characters may have different names and different characters may have the same name. To circumvent this problem, DeltaAccess allows linking of several projects to a common master character definition (project C & D in Fig. 6). The item definition and item description remain physically separate, which simplifies project management. If some characters of the master character definition are inappropriate for a given linked project, they can be made invisible in the linked project, creating a character subset.

Finally, it is possible to keep all data in a single master project, and create child projects that link to the character definition as well as to the item definition and description of the master project (project E & F in Fig. 6). The term 'project' is used here for a virtual data set, which takes the place of a physical data set in the view of the user. The characters and items visible in a linked child project can be restricted, creating character or item subsets. A static one-way filter facility is available in conventional DELTA programs in the form of the include/exclude characters/items directives. These directives can be used during analysis and reporting, but not during editing. DeltaAccess extends this concept using dynamic views, which allow the concurrent editing of multiple subset views of a project. A subset project can restrict both the characters and the items visible (see Fig. 7). These subset projects can be named and are handled identically to full projects. In practice, a researcher does not need to know whether she or he is working on a subset or on a full project. Multiple dynamic subset views contain identical data, not a copy of these data. Thus, if an item description is changed in one subset view, the changes will be directly visible in

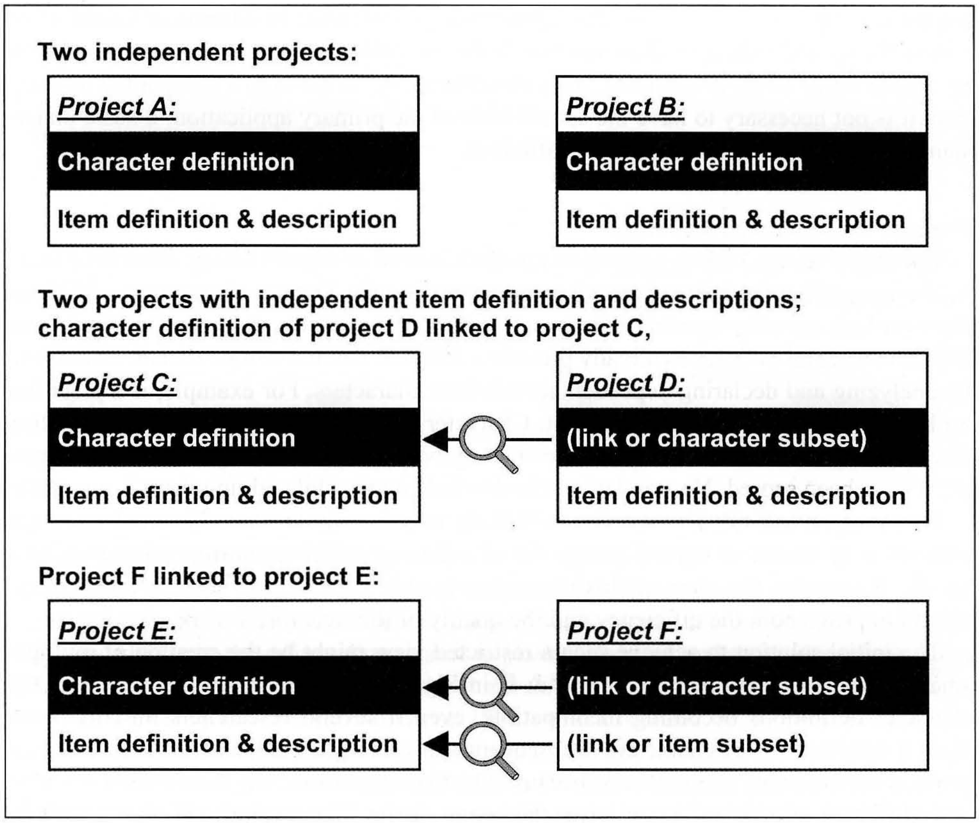
Two independent projects:
Project A:
Character definition
Item definition & description
Project B:
Character definition
Item definition & description
Two projects with independent item definition and descriptions; character definition of project D linked to project C,
Project C:
Character definition
Item definition & description
Project D:
(link or character subset)
Item definition & description
Project F linked to project E:
Project E:
Character definition
Item definition & description
Project F:
(link or character subset)
(link or item subset)

Fig. 6. Schematic diagram of possible relationships between projects. Projects can be entirely independent (project A and B), or linked to another project (projects D and F). Either both the character and the item definition (project E and F) or the character definition alone (project C and D) may be linked. In the latter case (project D) the item definition and description remain physically independent of the project with which it shares a common character definition. Optionally, a link may contain a restriction clause (symbolized by a looking glass), creating a subset of the original project.

all other views currently open. Dynamic views are an extension of the multi-user capabilities of a database software.

Finally, DeltaAccess offers a choice between static and dynamic item subset conditions. A static condition is a list of item numbers, just like it is used in a DELTA include/exclude items directive. In contrast, a dynamic subset condition is based on the item description data itself. For example, a subset may be defined containing all items which are found on a certain substrate. Whenever such a subset project is opened, the condition is evaluated and the presence of new or updated items is determined based on this condition.

Low-level data capture and summarizing data

Many scientists using DELTA compatible software still follow the classical workflow paradigm when they prepare a taxonomical monograph. First they record their observations

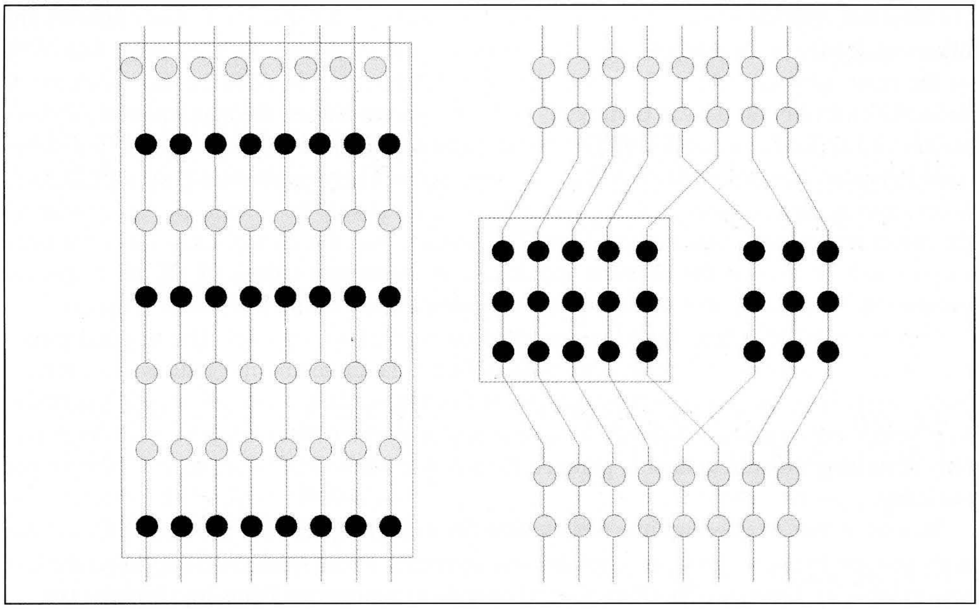

Fig. 7. Schematic diagram of a subset view to a project. The full project view on the left side contains all characters (vertical lines) and items (horizontal rows of dots). In the subset on the right side the view (symbolized by the rectangle) is restricted to selected characters as well as selected items (black dots).

on the selected specimen and the data extracted from literature references by some means, e.g. on index cards. They then summarize these data into computerized descriptions for each species. The problem with this procedure is that the descriptions do not carry any information about which part of a description was obtained from which source. If later a specimen turns out to carry a misapplied name, or, if in the course of the revision a literature reference is recognized to be not trustworthy, it is very difficult to correct the item descriptions. Usually the complete description must then be revised from the original sources.

It is obvious that already the original data capture should be performed in a way that allows it to be automatically converted into an item description summarizing all the information about a taxon. Low-level recording of original data and a summarize feature are not necessarily restricted to databases. However, one problem with conventional DELTA compatible software is that the unstructured item notes must be used to record to which specimen or literature reference an observation belongs. Descriptive databases usually offer extensions to the DELTA standard to store this information more appropriately. The direct links to other database subsystems (specimen/literature references) and the organized data storage make low-level data capture much more practical in a database. A summarize option is available, e. g., in PANDORA and DeltaAccess.

Table **1** gives a hypothetical example of the use of a summarize function for two species with two observations each. The primary item description data may be linked to a specimen, a literature reference or both. These data are then summarized into data describing the species as a whole. For categorical data (character 1 and 2 in table 1), the summarized data are identical with the union set of the states of all items of this species. If many species are observed, it may be desirable to include only states that are present in, e. g., more than 99% of the items of that species. For numerical data (character 3 in table 1), the summary is defined as the minimum of all minimum- and lower-range-values, the maximum of all maximum- and upper range values, and the mean of all mean-values. Species 2 in table 1 illustrates some of the additional problems that may occur when summarizing numerical data. If no mean is present in some items, an artificial mean should be calculated as the mean of the lower and upper range values. Also, it is possible that the normal range of some item could reach or exceed the absolute maximum- or minimum-values of all other species together, in which case the maximum- or minimum-values would have to be dropped.

Ideally, numerical data should be recorded on an even lower level. The original measurements or counts should be directly written into a database and the statistical summary (min, mean, max, range) should be calculated from these data. Since often only the statistical values are available, it should be also possible to enter them directly. Such functionality regarding numerical data is planned for a future version of DeltaAccess, but not yet available.

Besides summarizing multiple observation for a single species, it is possible to use the same process to create genus descriptions as a summary of the species descriptions (see last row in table 1). DeltaAccess offers several options to summarize items into higher taxa.

Data security

Some level of data security is required in almost any database. Even insensitive data, where neither legal, privacy, nor copyright or patent issues apply, must be protected from untrained users who may accidentally change or delete data. Several levels of security can be distinguished:

- *File-level security*, e. g. password protection of an entire database. Everybody knowing the password can access the database. This simplest security level can be turned on in a single step within DeltaAccess.

Table 1. Example of low-level data recording followed by a summarizing analysis. Character 1 and 3 are categorical, character 3 is a numerical character.

		Character:	— 1 —		— 2 —					— 3 —		
Item name	**Specimen** (accession #)	**Literature reference**	state 1	state 2	state 1	state 2	state 3	min	lower	mean	upper	max
Genus spec-1	B 123456		■		■			(2)	3	**4**	6	
Genus spec-1	M 3333 (type)	Author (1888)	■			■		(1)	2	**6**		(7)
Genus spec-2	G.H. 97-511			■				(3)	4		8	(9)
Genus spec-2		Author (1998)	■	■			■		2	**8**	9	
Summarized species descriptions												
Genus spec-1			■		■	■		(1)	2	**5**	6	(7)
Genus spec-2			■	■			■		2	**7**	9	
Summarized genus description												
Genus			■	■	■	■	■	(1)	2	**6**	9	

- *Network-based user- and group-level security*, i. e. validated logon to a network. The availability of this security feature depends on the network, not on the database software used. The simple password of the first level is replaced by a combination of user name and password. The entire database file is either available or unavailable.
- *Database-based user- and group-level security*, i. e. validated logon to the database. The advantage over the previous level is that differentiated permissions to objects within a database can be granted. Different users are allowed to do different things. In DeltaAccess, permissions can be given to tables, queries, etc. Thus some users can be granted read-only access to one project, but read and write access to another project. It is also possible to allow a group of users to edit item descriptions, but not the character definition.
- *Record- and attribute-level security within database.* In addition to the previous level this level allows to restrict access of a user or group to certain records (e. g. to all species of one genus) or to a subset of the attributes in a table. In DeltaAccess, a simple implementation of this security level is possible using a combination of item and character subsets (views) discussed in the previous chapter. Watertight record- and attribute-level security is also possible (using queries with 'owner permission'), but much more difficult to implement and manage. Yet, even with relaxed security, the simple editing restrictions using subsets prohibit accidental changes.

Choosing the right security strategy is not always easy. The higher security levels offer increased management options and flexibility, but the setup and management effort increases as well. No conventional DELTA compatible software package offers security features of it own. Thus, the application-independent, network-based security is the only security option available. In contrast, all major descriptor database applications offer one or more of the higher security options.

Distributed and replicated data

The following topic is discussed in more depth, because database replication is a relatively new feature, which can be essential when designing an international collaborative project. Distributed and replicated databases are used or planned, e. g., in IPNI (The Plant Names Project 1999) or the "Mediterranean lichens on-line" project (Grube & Nimis 1997).

While most databases are designed to work with multiple users in a local area network (LAN), only some databases can also operate in wide area networks like the internet. Some databases treat the internet as an extension of a LAN. The clients communicate directly with a single central database server, which may be located anywhere in the internet. Often standard WWW-browsers can be used as clients, which is a huge advantage. The disadvantages of this method are that the internet connection must have a high availability, and reliability. No interaction is possible if the internet connection is temporarily unavailable. For a secure environment it is necessary to encrypt all data traffic, which further slows down the operation. Records edited over the internet are usually not locked, creating possible conflicts when multiple users edit the same record.

An alternative is to distribute the data amongst multiple replicated database servers in multiple LANs (see Fig. 8). Each database server contains a copy of the entire data set or

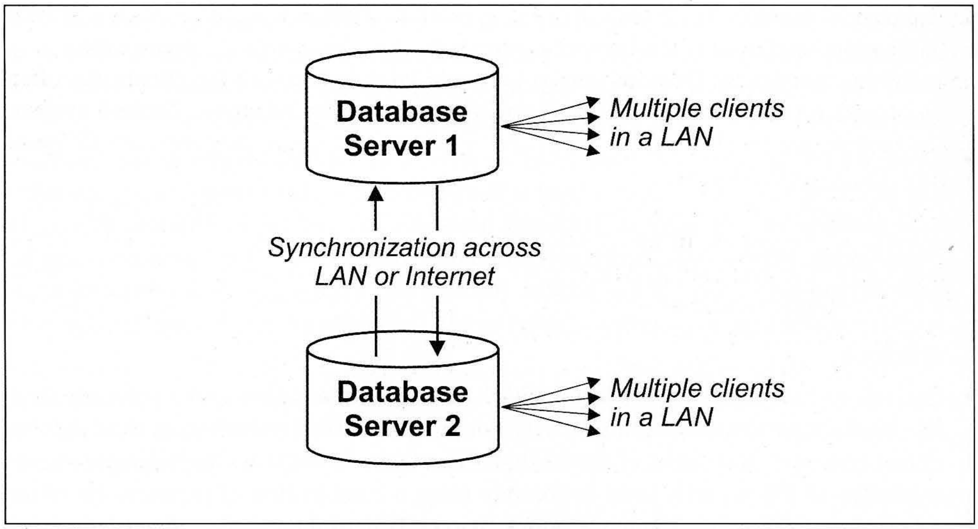
Database Server 1
Multiple clients in a LAN
Synchronization across LAN or Internet
Database Server 2
Multiple clients in a LAN

Fig. 8. Distributed data and database replication. Each of several database servers hosts the complete database and provides access to multiple local clients. If data are changed on one database server, the data are transferred to the other servers during the regular synchronization between the servers. The connection between the servers may be non permanent and have a relatively low bandwidth.

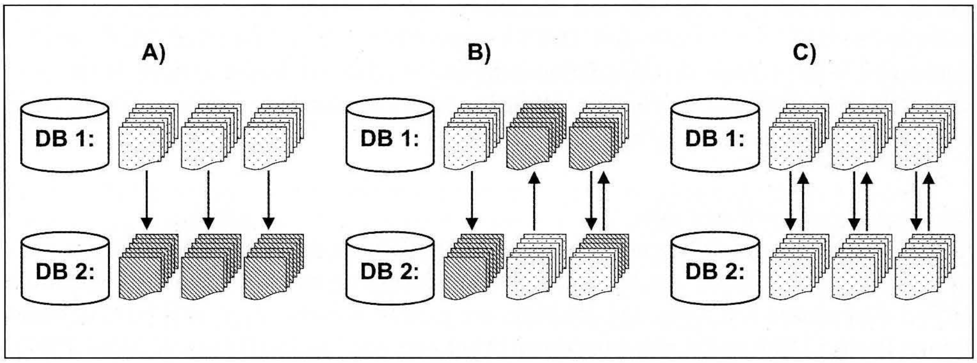

Fig. 9. Database replication strategies. **A)** Data warehousing: one or several database servers allow updates of the data (dotted tables), while other carry read-only copies (hatched tables). **B)** Distributed updating rights: Each database server is updating master for parts of the data, which are available as read-only copies on the other database servers. The updating master rights can be given for whole tables (left two tables), records within tables (right table), or attributes of a tables (not shown). **C)** Fully replicated database with balanced updating rights.

the parts relevant to the local clients. The clients access their local database server over a LAN and only the database servers communicate with each other across the internet. During regular intervals, the servers compare the changes made in the local copy of the database and synchronize the changes with each other. Synchronization may occur immediately ("as soon as possible"), scheduled at a certain time (e. g. overnight), or it may be under manual control. Since only periodic synchronizations are required, such a system remains functional even if the internet connection is available only at certain times, e. g. in a dial-up connection. Other advantages are that local traffic with the clients can often remain unencrypted, because the LAN itself is protected, e.g. through a firewall system. Since records are locally locked, multi-user conflicts can occur only between different servers in the replication set.

Several replication strategies are commonly used. They differ in which records can be updated at a given point. In first strategy (Fig. 9, A) only a single master database server allows users to update data (DB1), but users can access read-only copies of these data on one or several other database servers (DB2). The replication master automatically updates the data on these servers. This "data warehousing solution" is adequate if the second server is used for analytical or publishing purposes. In biological descriptive databases this strategy could, e.g., be used to distribute the load of identifications or queries coming from the internet. A second server (DB2) could handle this type of read-only operations and will be automatically updated during synchronization if updates occur on the primary database server (DB1).

The second strategy (Fig. 9, B) is similar to the first in that for each object (e.g. a table in a database) only a single server allows users to update the date. It differs in that different database servers may have the updating right to different objects in the database. Updating rights to can at least be granted on the table level, usually also for sets of records, or even for attributes (called 'partial replication'). The distributed updating strategy is applicable if the tasks in a project can be well divided in advance, but up-to-date information about other tasks is required. Examples would be inventory projects, where a common set of gazetteers is used, but the main transactions occur on adding data, e. g. specimen or organism names. Using a distributed and replicated database allows data retrieval of the most current data available, while the inventory work, which potentially lasts over many years, is still in progress.

Finally, it is possible to allow updates at any point within a replication set (Fig. 9, C). This strategy implies that the same record can be updated on multiple database servers. Such conflicts are detected by the database management software during synchronization. If transactions on existing records are rare compared to data entry and data lookup, or if the updates normally concern different records at different locations, it should be feasible to resolve these conflicts manually. Normally, this is the case with descriptive databases, since different researchers concentrate on different tasks and only occasionally add information or correct errors in other groups. The separation of tasks minimizes the chances of update collisions, even if synchronization is relatively infrequent.

Database replication is especially useful if data are to be edited in collaborative projects where a permanent network connection is not feasible or not desirable. Frequently encountered situations would be where a single computer is connected only by an expensive dial-up connection, or where a notebook is only occasionally connected to a LAN. With DeltaAccess, a project can be changed into a replicated project using the standard database replication functionality of Microsoft Access. All replication strategies discussed above can be implemented. Once this is done, a copy of the project can be carried on a notebook computer, edited, and later synchronized with a master copy of the project database on another computer.

Conclusions

A descriptor database subsystem like can provide the infrastructure for large-scale collaborative projects. Examples relevant to Mediterranean botanists are:

Interactive, synoptic keys to European and Mediterranean flowering plants
as part of the existing efforts to implement a continually updated version of the Flora of that region (Carine & al. 2000, Euro+Med PlantBase 2000). This project has recently started; for its taxonomic core it will use the PANDORA database system discussed above. Note that using descriptive databases, vegetative features can be freely integrated into interactive identifications (i.e. no distinction between keys for flowering and vegetative material is necessary).

Interactive, synoptic keys to the Lichens of Europe and the Mediterranean
based on the LIAS project (Rambold 1997, Rambold & Triebel 1997-2001). The LIAS project is already far advanced; data for generic identification and species of some genera have already been collected for several years using conventional DELTA compatible programs. DeltaAccess has been used in this project since 1997.

Index to plant parasitic fungi of Europe and the Mediterranean
This project should include a host-pathogen index (which pathogen on a host species or closely related species...), geographical distribution data (recorded in which country), and basic morphological characters. Each observation should be tied either to a literature reference or to a specimen. The final database should allow interactive identification using selected characters (including the geographical distribution and the host range). The German GLOPP project (Hagedorn 2000, Hagedorn & al. 2000) currently develops the basis for a full size European project.

Such projects require the collaborative effort of many researchers in many European countries. This can be facilitated by DeltaAccess. Many important features to support such projects are available (like interactive identification and HTML editing forms to work on the internet), some are under development (integration of illustrations and other resources, activation of links to other database subsystems, a full auditing/logging facility, etc.). DeltaAccess is being constantly improved; over the last year minor updates occurred approximately every three months. Even more important, the source code for DeltaAccess is supplied to the public under a General Public License, so that other workers can improve the software — in cooperation with the original author or in independent projects. DeltaAccess can be downloaded from http://www.bgbm.fu-berlin.de/projects/DeltaAccess.

Acknowledgements

I am grateful to W. Berendsohn, D. Maltais, G. Rambold, and M. Scholler for constructive criticism of this article.

References (including *electronic publications)

Berendsohn, W. G. 1995: The concept of "potential taxa" in databases. — Taxon **44:** 207-212.
*Berendsohn, W. G., Anagnostopoulos, A., Hagedorn, G., Jakupovic, J., Nimis, P. L. & Valdés, B.

1996: The CDEFD information model for biological collections. (Draft in HTML and WinWord 6 format). — http://www.bgbm.fu-berlin.de/CDEFD/CollectionModel/cdefd.htm.

— , —, —, —, —, —, Güntsch, A., Pankhurst, R. J. & White, R. J. 1999: A comprehensive reference model for biological collections and surveys. — Taxon **48:** 511-562. [preprint under http://www.bgbm.fu-berlin.de/ biodivinf/docs/CollectionModel/]

*BG-BASE Inc. 1997-99 (last accessed 31 Jan. 2001): BG-BASE™ Collection management software. — http://www.rbge.org.uk/BG-BASE/.

Carine, M., Heywood, V. & Jury, S. 2000: Euro+Med PlantBase: a new Euro-Mediterranean Initiative in Plant Systematics. OPTIMA Newsletter **25**: 21-23

*Centre for Pest Information Technology and Transfer 2001: LucID. — http://www.lucidcentral.com/.

*Cross, N. 1998: DeltaAccess Perl. — http://www.herbaria.harvard.edu /computerlab/ web_keys/delta_access_perl.html.

Dallwitz, M. J. 1980: A general system for coding taxonomic descriptions. — Taxon **29:** 41-46.

— 1993: DELTA and Intkey. — Pp. 287-296 in: Fortuner, R. (ed.) Advances in computer methods for systematic biology. — Baltimore, USA.

— , Paine, T. A. & Zurcher, E. J. 1995: User's Guide to the DELTA System: A General System for Processing Taxonomic Descriptions. Edition 4.02 (May). — Canberra.

*Dallwitz, M. J.; Paine, T. A. & Zurcher, E. J. 1999 (18 Jan., last accessed 31 Jan. 2001): Proposed new features for the DELTA System — http://biodiversity.uno.edu/delta/standard/proposal.exe.

*Euro+Med PlantBase 2000: Euro+Med PlantBase Newsletter 1, October 2000. — http://www.euromed.org.uk/newsletters/newsletter1.pdf.

*Free Software Foundation 2000 (31. July, last accessed 31 Jan. 2001): GNU General Public License — http://www.gnu.org/copyleft/gpl.html.

*Findling, A. 1998 (30.Sep., last accessed 31 Jan. 2001): DAWI - Ein Web-Interface zu DeltaAccess. — http://www.axel-findling.de/programs/dawi/.

Grube, M. & Nimis, P. L. 1997: Mediterranean lichens on-line. — Taxon **46:** 487-493.

*Gouda, E. 1999 (May, last accessed 31 Jan. 2001): Taxasoft (DELTA editor). — http://botu07.bio.uu.nl/gouda/taxasoft.htm.

*Hagedorn, G. 1997-2001. DeltaAccess – a SQL interface to DELTA (Description Language for Taxonomy), implemented in Microsoft Access. User Guide and Documentation. — www.bgbm.fu-berlin.de /Projects/DeltaAccess/.

— 2000: Globales Informationssystem zur Biodiversität pflanzenpathogener Pilze (GLOPP). — Nachrichtenbl. Deutsch. Pflanzenschutzd. **52** (6): 149.

— , Deml, G., Burhenne, M., Guerrero Cartin, O. M., Gräfenhan, T. & Weiss, M. 2000: Synoptische, computergestützte Identifizierung von Pflanzenpathogenen. — http://www.glopp.net/GLOPP/Presentations/PflSchutzTg2000/PflSchutzTg2000.htm.

— & Rambold, G. 2000: A method to establish and revise descriptive data sets over the internet. — Taxon **49:** 517-528.

Maddison, D. R., Swofford, D. L. & Maddison, W. P. 1997: NEXUS: An extensible file format for systematic information. — Syst. Biol. **46:** 590-621.

Pankhurst, R. J. 1991: Practical Taxonomic Computing. — Cambridge, UK.

— 1993: Taxonomic databases: the PANDORA system. — pp. 229-240 in: Fortuner, R. (ed.) Advances in computer methods for systematic biology. — Baltimore, USA.

— & Pullan, M. R. 1996: DELTA in PANDORA. — DELTA Newsletter **12:** 16-20.

*Pankhurst, R. J. 1998a: Pankey. A suite of key generation and identification programs. — http:// www.rbge.org.uk /pankey.html.

*— 1998b: The PANDORA taxonomic database system. — http://www.rbge.org.uk/research/pandora.home.

Rambold, G. 1997: LIAS – the concept of an identification system for lichenized and lichenicolous ascomycetes. — Pp. 67-72 in: Türk, R. & Zorer, R. (ed.), Progress and problems in lichenology in the Ninties -IAL 3. - Biblioth. Lichenol. **68**.

— & Hagedorn, G. 1998: The distribution of selected diagnostic characters in the *Lecanorales*. — Lichenologist **30:** 473-487.

*Rambold, G. & Triebel, D. 1997-2001: LIAS: Lichenized and Lichenicolous *Ascomycetes*. — http://www.mycology.net/lias/.

Swofford, D. L. 1990: PAUP. Phylogenetic analysis using parsimony. — Champaign, Illinois.

TDWG 2000: International Working Group on Taxonomic Databases. TDWG standards. — http://www.tdwg.org/standrds.html [last accessed 31 Jan. 2001].

The Plant Names Project 1999 (last accessed 31 Jan. 2001): International Plant Names Index. — http://www.ipni.org.

*Watson, L. & Dallwitz, M. J. 1992 (and onwards): Grass genera of the world: Descriptions, illustrations, identification, and information retrieval; including synonyms, morphology, anatomy, physiology, phytochemistry, cytology, classification, pathogens, world and local distribution, and references. Version: 30 April 1998. — http://biodiversity.uno.edu/delta/ (character list: .../grass/www/chars.htm).

Watson, L. & Dallwitz, M. J. 1994: The grass genera of the world. 2nd ed. — Wallingford, UK.

White, R. J., Allkin, R. & Winfield, P. J. 1993: Systematic databases: The BAOBAB design and the ALICE system. In: Fortuner, R. (ed.) Advances in computer methods for systematic biology. John Hopkins Univ. Press. Baltimore, USA Pp. 297-311.

White, I. M. & Sandlant, G. R. 1998: Computerised insect identification: a comparison of differing approaches and problems. — Pp. 261-271 in: Bridge, P., Jefferies, P., Morse, D. R., & Scott, P. R. (ed) Information Technology, Plant Pathology and Biodiversity, Wallingford, UK.

Address of the author:
Gregor Hagedorn, Institut für Pflanzenvirologie, Mikrobiologie und biologische Sicherheit, Biologische Bundesanstalt für Land- und Forstwirtschaft, Königin-Luise Str. 19, 14195 Berlin, Germany.