

RADAR, a Relational Archaeobotanical Database for Advanced Research

Henk van Haaster¹ and Otto Brinkkemper²

¹Instituut voor Pre- en Protohistorische Archeologie, Universiteit van Amsterdam, Nieuwe Prinsengracht 130, 1018 VZ Amsterdam, The Netherlands

²Instituut voor Prehistorie, Rijksuniversiteit Leiden, P.O. Box 9515, 2300 RA Leiden, The Netherlands

Received August 22, 1994 / Accepted January 24, 1995

Abstract. The structure of the Dutch Relational Archaeobotanical Database (RADAR) is presented. RADAR is a rather compact archaeobotanical database that is controlled centrally, but can be distributed to individual scientists. For this reason RADAR contains only the most important archaeobotanical data. For detailed archaeological, botanical and regional palaeoenvironmental information, links can be established with the national archaeological database (ARCHIS), the national botanical database (BBR) and the European Pollen Database (EPD). The software used for manipulation of the database is PARADOX for reasons of its highly visible nature, its control facilities for data entry and the ease of importing and exporting data from and to many other programs. The potential of the database is demonstrated with query examples.

Key words: Archaeobotany – Relational database – The Netherlands – PARADOX

Introduction

Archaeobotany, the study of plant remains from archaeological contexts, is an expanding branch of research in archaeology. The epoch in which questions concerning the food plants in any archaeological period and locality dominated the scope of research has been left behind for several decades. The natural palaeoenvironment surrounding archaeological sites, its influence on human habitation and vice versa are now commonly addressed subjects. As a result, not only remains of plants with economic importance are identified and recorded, but those of wild plants as well. In addition, more and more other remains than the "traditional" seeds and fruits are being recognized. Some taxa are now being identified from their pollen, epidermal fragments, bud scales etc. even though their fruits and seeds have a little chance of being

preserved. For example, pollen analysis of latrine deposits has added many important taxa (herbs, spices, vegetables) to the species lists of latrine studies; evidence which does not belong in pollen diagrams as it is part of the palaeo-ethnobotanical record (see Greig 1994).

As a result, a considerable body of data is being generated with a great scientific potential not only for archaeology but for other fields of research like plant history and biogeography as well. In view of this, it is of great importance to facilitate the recovery of archaeobotanical data that are present in hundreds of reports, many of which are only known to a small group of scientists because they may be internal laboratory reports or appendices to archaeological publications.

A system that makes storage and recovery of all archaeobotanical data more efficient becomes more and more desirable. With increasingly fast personal computers available in almost every laboratory today, the idea of a computerized archaeobotanical database comes into view. One major problem in the design of an archaeobotanical database is that it is virtually impossible to put all the relevant information into one single database. The ultimate archaeobotanical database should not only contain a list of identified taxa with sample numbers and quantification, but it should also contain information about sample volume, mesh sizes, feature type, date of the context, site type and context, topographical information and bibliographical information of the report(s) that refer(s) to a site.

Putting all this information into one single database would make it clumsy and inefficient. For example, for every new plant record that one wished to add to the database one would have to duplicate all the other information about the report, site, feature and sample that had already been put in the database when the first plant record of the same sample was entered.

A better solution would be to store the information in more than one database. In this case a problem arises when information is needed from more than one table simultaneously, because most file managers can only access one database at the time. In this case a *relational database* is needed.

The powers of a relational database

A relational database is a system where separate databases are linked by means of common fields. The common fields contain identical information which can be used to match one database to another. Because the common fields are the key to the retrieval of information from a relational database, they are called *key fields* in relational database terminology.

In Fig. 1 an example is given of a relational database. A table with sample information is related to a table with plant identifications from the same samples.

Samples table

Sample nr.	Sample code	Sample type	Weight/volume	Remarks
534	dbkj1834	gba	250 g	dat
535	dbkj1301	pol	?	
536	adnd867	gba	4 l	
537	adnd347	flot	150 g	

Plants table

Sample nr.	Plant name	Id.lev.	Part	Pres. mode	Quant.	Remark
534	Avena sativa	spec	flb	c	6	
534	Avena sativa	cfspec	see	m	36	id?
534	Poa annua	spec	see	w	2	
534	Taraxacum	cfgen	see	w	1	
535	Avena	type	pol	w	24	
536	Vicia faba	spec	see	c	3	

Fig. 1. Two tables in a relational archaeobotanical database. gba = sample for general biological analysis, pol = sample for pollen analysis, flot = flotation sample, flb = flower base, see = seed (s.l.), pol (in Part field) = pollen grain, c = carbonized, w = waterlogged, m = mineralized. The Remarks field serves for comments from the national database manager: dat = more details about the dating are published in the original report, id? = doubtful identification (according to the national database manager)

Both databases have the field "Sample nr" in common. In this example the Sample nr fields serve as key fields. By searching for the sample number in the Samples table and matching this to the sample number in the Plants table, a database manager can determine which taxa were found in what type of sample. A database manager that can draw information from more than one database at the same time, is called a *relational database manager*. With the help of such software the user is able to access more than one related database, while he or she has the impression of only working with one.

Tomlinson (1992) was the first to introduce a relational database for archaeobotanical data from Great Britain and Ireland. Her Archaeobotanical Computer Database (ABCD) has been set up on a mainframe computer. The databases were created and manipulated with the help of Structured Query Language (SQL). No use was made of a relational database manager. According to

the author SQL is at its basic level a simple database language. However, the query examples in the above mentioned article are rather complicated. Most queries are strings of complicated commands alternated by brackets, commas, semi-colons etc., that have to follow a precise syntax and are difficult for the average archaeobotanist to understand. The author is aware of that when she states that '...for other users and for improving data checking and integrity systems, an easy to use front-end will be necessary.'. Since no single database package emerged as the most often used in Great Britain and Ireland, it was decided to keep the database centrally at the Environmental Archaeology Unit in York. The easy to use front-end would no longer be necessary in this case as the database is maintained and controlled centrally by a computer specialist.

Maintenance and updating of a national database can indeed best be organized centrally by a specialist. However, in the ABCD concept the queries will also have to be carried out centrally by the database specialist. In our opinion such a concept does not encourage the research of the individual archaeobotanist as the questions will have to be sent to the National Database Manager, who translates them into database queries. We feel that a system like this is not the most inspiring.

History teaches that most progress is made when individual scientists have the freedom to "play" with their data, in a process of trial and error where each question evokes another. In other words: the enormous scientific potential of the Archaeobotanical Computer Database in its present form¹ is not utilized to its full extent.

RADAR, the national archaeobotanical database for the Netherlands

Design and basic structure

In our concept a national database should be maintained and controlled centrally, but individual scientists should have easy access to it. Therefore, the database will be distributed to those who provide data. All data in the database are unrestricted. Those who provide data make the selection which of their data will be included. This concept of a "mobile national database" implies that individual scientists should be able to perform the queries they want, without typing strings of "computer language" and without knowledge of computer programming.

One major implication of the perception of a mobile national database is that it tends to get too bulky to be distributed as a whole. This implies that the database should be as compact as possible. It should only contain the most important archaeobotanical data. Data which are not directly archaeobotanical (detailed archaeological information, ecological plant data) should be held in other (inter)national databases. However, in itself, the

¹The author is currently working on methods to facilitate the distribution of the database (Tomlinson, pers. comm.).

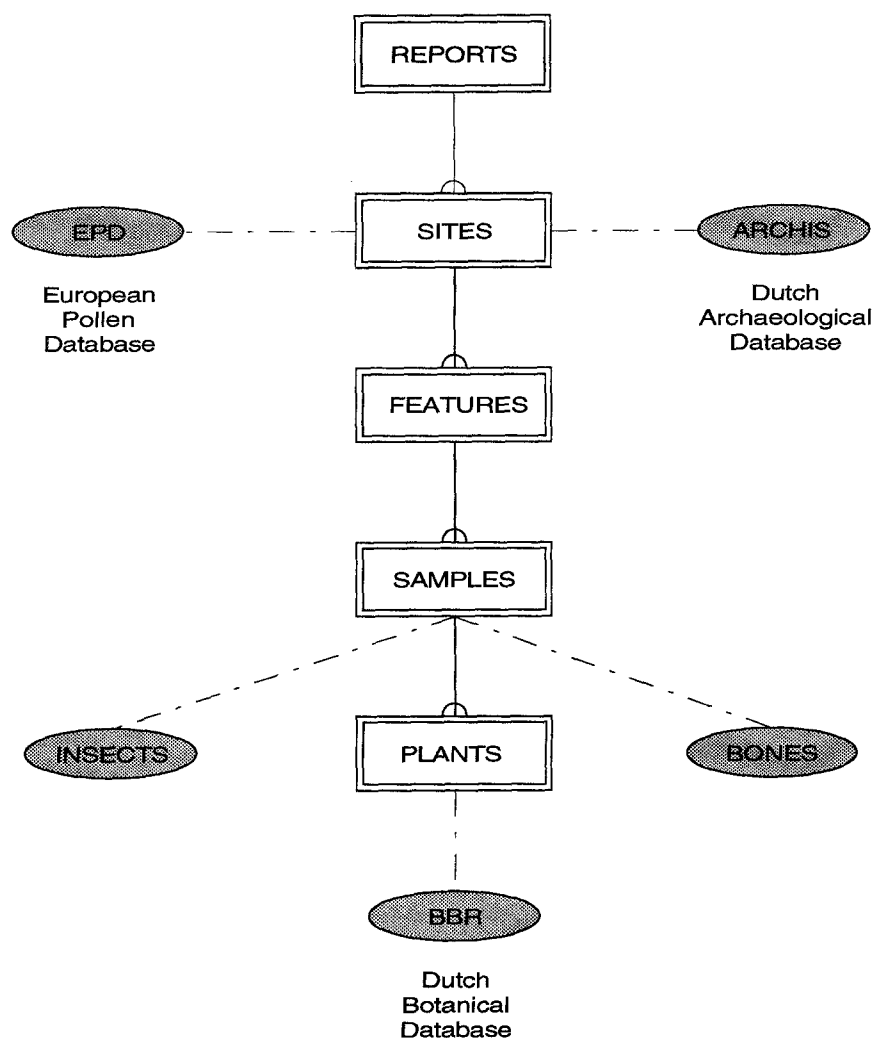


Fig. 2. Basic structure of RADAR with optional links to other (inter)national databases

archaeobotanical database should offer the possibility of giving answers to the most "common" archaeobotanical questions (e.g. find occurrences of a certain taxon with information on dating, context, sample type, site type, bibliography, location, etc.). It should, however, be possible to link RADAR to other relevant databases.

With the above mentioned conditions we decided to design an archaeobotanical database with five tables which are shown in Fig. 2. The tables (databases) are represented by boxes. The lines indicate the relationships between the tables. Boxes drawn with shaded boxes represent other (inter)national databases. The dotted lines indicate the optional relationships of those databases with RADAR.

In designing RADAR we assumed that each report deals with one or (occasionally) more sites, that each site consists of one or more features (wells, postholes, pits, etc.), that from each feature one or more samples are taken and that in each sample one or more plant remains are present. With this assumption the data can be arranged in

five hierarchically organized databases with the Reports table in top position. In this way even archaeobotanists not familiar with database management software can understand how the relationships between the databases are established. We feel that this will reduce the chance of wrong results from "home made" queries. However, in some cases, the data cannot be framed in this straightforward way. This is the case if more than one archaeobotanical report has been published about a certain site. This happens if for instance the different cultural periods within a site are investigated and published by different authors. Also, sites are sometimes revisited after a few years, resulting in more than one archaeobotanical report per site. Likewise, the basic structure of RADAR assumes that features have only one dating range. Occasionally, however, more than one dating range is obtained from one feature. This can be the case with features showing stratigraphy. As these circumstances occur only very occasionally, the most efficient way to cope with it is to enter these sites or features more than once in the corresponding tables with different numbers.

RADAR tables and their internal structure

In Fig. 3, a list of the RADAR tables with the field names used within each table is given. The field names represent table columns with the different categories of information. The so-called key fields, which are used to link the separate tables, are marked with asterisks.

REPORTS

- *report nr*
- report title
- author(s)
- editor(s)
- title of book/journal
- year of publication
- status (published, internal report, unpublished)

SITES

- *report nr*
- *site nr*
- ARCHIS identification code
- X-coordinate (according to national grid system)
- Y-coordinate (according to national grid system)
- province
- site name
- site type (e.g. cemetery, castellum, villa, sanctuary, monastery)
- site context (e.g. rural, urban, military.)
- cultural period (of the site as a whole)
- begin date (of whole site)
- end date (of whole site)
- remarks

FEATURES

- *site nr*
- *feature nr*
- feature type (e.g. posthole, dung layer, pit, silo, ditch, well, cesspit)
- begin date
- end date
- local phase (to indicate relative age when exact age is not known)
- remarks

SAMPLES

- *feature nr*
- *sample nr*
- sample code (authentic sample number to facilitate checking of data in original publication)
- sample type (according to national sampling and recovery methodology)
- sample volume/weight
- remarks

PLANTS

- *sample nr*
- *plant code* (according to national Botanical Database BBR)
- *plant name*
- *identification level* (e.g. species, genus, family, type, cf)
- *part preserved*
- *preservation mode* (waterlogged, carbonized, mineralized, subrecent)
- quantification
- remarks

Some notes on the internal structure of the tables

As stated above, common fields are used to link the separate tables. The common fields serve as keys for the retrieval of information from related tables. Like door keys, key fields in a database must contain unique information. In other words, a database can never contain two or more records with identical keys. For example, if there were two features with the same feature number on a given site, a database manager could never detect the difference between both features. Similarly, a site number should occur only once in the Sites table, etc. Note that a site number may occur more than once in the Features table. It is the combination of site number and feature number which serves as unique key in the Features table. In some cases a combination of two fields is not sufficient to make a key unique. This is the case in the Plants table. If both carbonized and waterlogged remains of a certain taxon are found in the same sample, not only the sample number and the taxon name will have to be duplicated in the next record. In this example the combination of the sample number and the taxon name does not serve as a unique identifier for that record. Therefore the fields Identification level, Part preserved and Preservation mode must be included in the key as well (compare the first two records in Fig. 1). Although the latter key fields are not used to link the Plants table to the Samples table (like the common field Sample nr), they ensure that the total key in each record is unique.

It is important to keep key fields as short as possible, as this is of great influence on the speed of answering queries from linked tables. This is the reason why we decided not to use the field with authentic sample codes as key field. Instead of this we use a key field with our own numerical sample numbers. This field can be kept much shorter than the field with authentic sample codes. The latter field can be used to facilitate the checking of data in the original publications.

It is also of great importance to use a uniform terminology for the attributes or values in a field. This means for instance that attributes or codes for site type and context, feature type, cultural period, sample type and plant parts and names will have to be standardized. *Polygonum*, *Fallopia* and *Bilderdykia convolvulus* for instance all refer to the same species and Roman Iron Age and Roman Period to the same cultural period. We feel that this paper is not the right place to describe all the attributes or codes that are necessary in a national archaeobotanical database. In many cases existing standardized terminology can be used. For the plant taxonomy, we follow the Dutch flora (van der Meijden 1990), which follows *Flora Europaea*. Tomlinson (1992) gives valuable suggestions for many attributes. For the archaeological information in RADAR, we used the standardized terminology developed for ARCHIS, the national archaeological database in the Netherlands. Sample types are according to a national sampling and recovery methodology.

Fig. 3. The different tables of RADAR and their structure. ARCHIS is the Dutch national archaeological database

The Quantification field poses a special problem, as different authors use different ways of quantifying plant remains. Both absolute numbers and presence/absence or estimated amounts may be published. However, to take full advantage of the crosstab function (see below), the Quantification field should be numerical. For this reason, abundancy codes other than numbers will have to be transformed to numerical codes. In this way, even non-numerical quantifications can be incorporated in a crosstable. In the Remarks field of the Plants table, information about the method used for quantification is recorded.

A crosstable can also be used for numerical analyses. However, the possibilities for these analyses in database packages are limited. For elaborate statistical analyses, the data will have to be exported to other software. The export of data to formats that are accepted by statistical packages such as SPSS and CANOCO is dealt with below.

The X- and Y-coordinates can be exported to a Geographical Information System (GIS). A GIS can perform data analyses in relation to various kinds of geographical information and it can produce distribution maps.

A recurring problem during the entry of data from published reports is that not all the reports contain the necessary information. In many cases information about sample type, mesh size, or context type is not present in the report. Other problems are the use of obsolete taxonomic nomenclature or the lack of clarity concerning preservation mode or part preserved. In the Remarks field short codes are used to deal with these problems.

How RADAR is linked to other (inter)national databases

The possibilities for linking RADAR to other databases mainly depend on the availability of other databases in the relevant country. If specific archaeological information of a site is needed, the Sites table can be linked to an archaeological database. For the Netherlands, the archaeological information is provided by ARCHIS, the Dutch Archaeological Database (Brandt et al. 1992). In ARCHIS detailed archaeological information, dating method, depth below surface and reports about a site are recorded. ARCHIS also provides general information about the environmental setting of a site such as soil type, geological and geomorphological context. The link to ARCHIS is established by means of the ARCHIS identification code of a site.

If regional palaeoenvironmental information is needed, the U.T.M.-coordinates, which can be derived from the X- and Y-coordinates (see van Nieuwerkerken 1991), can be used to select palynological information from the European Pollen Database.

If botanical or ecological information about plants is needed, RADAR can be linked to the Dutch Botanical Database (BBR), which is also indirectly based on *Flora Europaea*. This database contains a large number of data about the taxonomy, plant geography, morphology, phenology, phytosociology, and autecology of all vascu-

lar plant species occurring in the Netherlands. The database is distributed by the Netherlands Central Bureau of Statistics (van Duuren 1993). Linking to the BBR is established by means of the official taxon names². If databases for other bioarchaeological data (insects, animal bones) are available, linking to RADAR is as indicated in Fig. 2.

All links to the above mentioned databases are optional. In itself RADAR answers the most common archaeobotanical questions.

The choice of PARADOX as database management package

On the software market are many products that offer relational database management for IBM-compatible computers, including dBASE IV, R:BASE, REVELATION, FOXPRO, PARADOX and others.

The aims of the Dutch archaeobotanical database make specific demands on the software. In the first place it should provide an easy to use front-end. It should be possible to perform queries without having to type strings of computer language. Furthermore, the software should have sufficient built-in safety warrants against inappropriate use by those who are not familiar with computer systems. Also it should be easy to import and export data from and to a great variety of other formats. An option to generate crosstables would be very useful in order to obtain tables that are ready for publication. In our view, PARADOX by far exceeds other software in these respects. PARADOX is a product of Borland International, Inc. All rights are reserved by this company.

Data manipulations

Data entry

The use of any database involves two main parts. Firstly, we have to deal with input of data and secondly we have to be able to retrieve selected data. Thanks to the ability of PARADOX to import data from a great variety of other database and spreadsheet packages, including QUATTRO or QUATTRO PRO, LOTUS, SYMPHONY, dBASE, PFS:FILE, REFLEX, VISICALC and ASCII, we often simply need to reorganize existing tables and import the data in RADAR.

Tables in "real" database format (e.g. REFLEX, dBASE etc.) can be imported directly into PARADOX, after which they can be added to an existing RADAR table.

The import of tables in spreadsheet formats demands special attention, as the structure of a spreadsheet is quite different from that of a database. In spreadsheets with archaeobotanical data, the first column usually con-

² In the near future, all plant names will be replaced by a numerical code, which corresponds to the taxon code in the BBR. This strongly decreases the size of the Plants table.

tains taxon names, while the other columns contain numbers or abundance codes. The column headings, in this example, are sample codes. Tables with a structure like this are called *crosstables*. In order to take full advantage of the capabilities of PARADOX, crosstables will have to be "uncrossed" i.e. converted into a "real" database. The process of uncrossing is shown in Fig. 4. Note that the Taxon and Sample code fields in this database should be reordered to ensure correct inclusion in RADAR.

spreadsheet (crosstable)

Taxon	Sample A	Sample B
Prunus	10	20
Rubus	30	0

"real" database

Taxon	Sample code	Quant.
Prunus	Sample A	10
Prunus	Sample B	20
Rubus	Sample A	30

Fig 4. Transformation of a spreadsheet to a real database

To convert spreadsheets with archaeobotanical data into real databases, we use a script called UNCROSS. A script is a combination of keystrokes and/or commands which are saved as a file. Once made, a menu option can be used to "play" the script again (see below). The script is written in PARADOX APPLICATION LANGUAGE (PAL).

After a spreadsheet is uncrossed it can be combined with an existing table. Note that the new table must have the same structure as the comparable existing RADAR table (compatible field types, arranged in the same order).

PARADOX offers a menu option called ADD, which makes it possible to combine new tables with existing ones. When new tables are added the referential integrity of the new combined table is checked. This means that PARADOX does not allow the entry of a new record with the same key value into an existing table. Records with identical (combinations of) key fields are placed in a temporary table called KEYVIOL. After the necessary adjustments have been made, the refused records can be added again.

Unfortunately, the information about the samples, features, sites and reports is usually not available in ready to use computerized formats. These data will have to be entered manually. New data can be entered directly into existing tables by editing the tables, but this is not the safest way to do it. While editing an existing table, there

is always the chance that unwanted changes are made. PARADOX offers a data entry choice that places new entries into a temporary table named ENTRY. After this temporary table has been checked for errors, it can be added to the existing table. In this case PARADOX again controls the referential integrity of the new combined table. So-called *key-violations* are placed automatically in a temporary KEYVIOL table.

A powerful feature of PARADOX is that new input in tables can be controlled by so-called *validity checks*. With help of this option, conditions can be defined that entries into fields must meet before they will be accepted. The EDIT menu option VALCHECK offers various ways of controlling data entry. One strong option in VALCHECK is the possibility of defining so-called *lookup tables*. A lookup table contains standardized values or names and can be used as a reference table. With a lookup table of valid taxon names, for example, all entries in the field Taxon name can be controlled. Entry of a taxon name is only accepted if a match for the entry is found in the lookup table with taxon names. Lookup tables are existing RADAR tables or can be other tables. In RADAR we use lookup tables with standardized names or values for taxon names (including standardized type names!), feature types, site types etc. Standardized taxon names are of the utmost importance in the retrieval of data.

New entry of key values can best be controlled when the entry is performed from the Reports table downwards. In this way new entries of key values in a lower RADAR table can be checked by defining a higher RADAR table as lookup table. For example, when new data must be entered in the Sites table, the report numbers in the Sites table are checked by defining the Reports table as lookup table for the entry of report numbers. Likewise, in the following Feature table, the Sites table is checked for the presence of the entered site numbers. In this way, a key value that is not present in a "higher" RADAR table can never be entered. This is of vital importance to ensure correct linking of the databases.

Another convenient option that VALCHECK offers is the possibility of setting a default (standard) value that PARADOX will place automatically in a particular field if nothing is entered. So the easily changeable default values for "part preserved" and "preservation mode" can be defined as "see" (seed or fruit) and "w" (waterlogged).

Query design

The main goal of a database of this kind is, of course, to get selected information out of it. In PARADOX, the basic step for retrieving data is to fill out a Query Form that is displayed on the screen when one or more tables are "asked" with the ASK menu option. Fig. 5 shows an empty Query Form that appears when a hypothetical Sites table is chosen with ASK.

The selection of data from this table comprises two simple, basic steps. The first step is to place checkmarks (✓) (in British English, ticks) in the fields that are desired in the answer. The second step is to place one or

SITES	Site nr	Site name	Province	Site type	Cultural period

Fig. 5. Example of an empty Query Form

SITES	Site nr	Site name	Province	Site type	Cultural period
	<input checked="" type="checkbox"/>		Utrecht	castle	<input checked="" type="checkbox"/>

Fig. 6. Example of a filled-in Query Form to select records from a hypothetical Sites table

REPORTS	Report nr	Report title	Author(s)	Year of publication
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> >=1985

SITES	Report nr	Site nr	Site name	Province	Site type	Cultural period
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	Utrecht	castle	<input checked="" type="checkbox"/>

Fig. 7. Example of partially filled-in Query Form for two related tables

more selection criteria in any desired field, if a specific selection of records is needed. Fig. 6 shows a filled-in Query Form for the selection of castle sites in the Dutch province of Utrecht. Checkmarks are placed in the fields Site nr, Site name and Cultural period. Once the Query Form has been filled in, the touch of one key (F2) is enough to perform the query.

The resulting ANSWER table would list all the castle sites from the province of Utrecht with the site name, site context, and cultural period. The ANSWER table is a temporary table that can be renamed (with easy menu option) if a permanent table with the selected data is needed. The Instant Report option (ALT-F7) gives a standard printed report of the query.

These steps are all that is needed to get selected information from a single database. In addition PARADOX offers a wide range of query symbols to facilitate queries that are more complicated, including operators used for calculations.

In most instances information is needed from more than one table simultaneously. In those situations the desired tables are chosen with the ASK option and a multiple Query Form is filled in. The only difference from a single Query Form is that the tables will have to be linked by means of their common fields. The links are established by means of so-called *example elements*. Two tables are linked by placing identical example elements in their common fields. Any string of characters or numbers can be used as example element (but no spaces or punctuation) as long as the example elements in the common fields of the two tables are the same. In PARADOX, example elements are placed with the Example key (F5).

As an example, Fig. 7 shows a workspace containing the filled-in Query Forms from the Reports and the Sites tables. Both tables are linked with the example element "rep". In this example a query is designed to obtain de-

REPORTS	Report nr	Report title	Author(s)	Year of publication
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

SITES	Report nr	Site nr	Site name	Province	Site type	Cultural period
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Utrecht	castle	<input checked="" type="checkbox"/>

FEATURES	Site nr	Feature nr	Feature type	Begin date	End date	Local phase
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	cesspit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

SAMPLES	Feature nr	Sample nr	Sample type	Sample volume/weight
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

PLANTS	Sample nr	Taxon name	Preservation mode	Part preserved
	<input checked="" type="checkbox"/>	Prunus avium	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. 8. Example of a partially filled-in Query Form of five linked tables. Note that this figure does not show the real RADAR tables. For reasons of simplicity some fields have been left out

REPORTS	Report nr	Report title	Author(s)	Year of publication
	rep1		✓	✓
	rep2	✓	✓	✓
	rep3	✓	✓	✓

SITES	Report nr	Site nr	Site name	Province	Site type	Cultural period
	rep1	site1	✓	✓	castle	✓
	rep2	site2	✓	✓	castle	✓
	rep3	site3	✓	✓	castle	✓

FEATURES	Site nr	Feature nr	Feature type	Begin date	End date
	site1	feat1	cesspit	✓	✓
	site2	feat2	cesspit	✓	✓
	site3	feat3	cesspit	✓	✓

SAMPLES	Feature nr	Sample nr	Sample type	Sample volume/weight
	feat1	samp1	✓	
	feat2	samp2	✓	
	feat3	samp3	✓	

PLANTS	Sample nr	Taxon name	Preservation mode	Part preserved
	samp1	Prunus avium	✓	✓
	samp2	Prunus avium	✓	✓
	samp3	Prunus avium	✓	✓

Fig. 9. Example of a Query Form of five linked RADAR tables, in which finds of *Prunus avium* in cesspits in castles within a specified date range (1500-1600 A.D.) are sought

tailed report information from the castle sites that were selected in Fig. 6. Because only reports are wanted that were published in or after 1985, the criterion ">=1985" is placed in the field Year of publication.

For most archaeobotanical queries, all five tables that make up RADAR will have to be linked. RADAR is designed in such a way that there can be no doubt about how to link the tables. The Query Forms are always arranged hierarchically, following the basic structure given in Fig. 2. Fig. 8 shows a multiple Query Form, designed to retrieve all finds of *Prunus avium* found in cesspits in castles, with information about report title, author, year of publication, site name, dating, sample type, preservation mode and the part preserved. Note that each set of two tables is linked by a unique example element.

When a selection of finds from a specific period is needed, a special Query Form has to be designed. This is because the desired dating range does not necessarily have to match the dating ranges of the features. If, for example, finds from the 16th century must be selected, we are also interested in finds that have an earlier begin date and an end date in the 16th century. This holds true for finds with a begin date in the 16th century and an end date after the 16th century as well. We are also interested in features with begin dates before and end dates after the 16th century. So the query has to be performed using three possible selection criteria for Begin date and End date. In PARADOX the extra search criteria are entered in additional rows in the Query Forms. In the examples above, all the checkmarks, selection criteria and example elements are entered in the first three rows of the Query Forms. In Fig. 9, an example is given of a query for all finds of *Prunus avium* from rural sites in the 16th century, with additional information about report title, author, year of publication, feature type, begin date, end date and sample type. Instead of typing the

whole name, the wildcards @ (any single character) and .. (any string of characters of variable length) are very useful here. Thus, Prunus.. would list finds of other *Prunus* species as well.

An important note about duplicate records

PARADOX does not include records in the answer table that it considers to be duplicate records. This is an important point to realize because, depending on the aim of the query, it may or may not give a deceptive answer. Records are considered duplicates when they contain the same values in all the fields in the answer table. Imagine a query, designed just to see at what sites a certain taxon is found, with some additional report information. More fields than just Taxon name, and a few fields in the Reports table are not supplied with checkmarks. The resulting answer table, in this example, would contain not more than one record of a taxon per site. However, there is a good chance that more records from the same site and taxon are present in RADAR, for example in different samples. All the possible other records in each site are considered to be duplicates. When extra fields, for instance Begin date and End date are checkmarked, more records per site may be placed in the answer table.

Recording and playing of scripts

The design of a query can be a time consuming process, especially when several related tables are involved. Fortunately, PARADOX offers the possibility of saving queries for later reuse. Once a query is designed and once it has been tested to see whether it gives the intended results, it can be saved as a Script. Scripts are, generally speaking, a combination of key strokes and/or commands that are saved in a file (in some other programs these are known as macros).

In RADAR we designed several scripts with many often used example elements, checkmarks and selection criteria already placed in the right fields. Variable selection criteria like taxon name, site type or context are left blank.

As shown in the example in Fig. 9, queries that look for the presence of any object within a certain period are rather time-consuming and complicated to design, because it is necessary to fill in three lines in the Query Forms. Also, the rather complicated dating selection criteria are easily forgotten. Since there are many cases where there will be the need to compare the results of a certain analysis with other finds from the same period, a Period Script is valuable. The script can be played back as many times as necessary, while only the dates or the taxon name (and possibly other criteria) have to be replaced. Scripts mentioned in this text can be obtained from the authors.

Export of selected data to other formats

The export of files from PARADOX to other database or spreadsheet formats is as easy as importing from those formats (see *Data entry*). Export is possible to eight formats which can be chosen from a menu (Tools/Export/Import), to QUATTRO or QUATTRO PRO, LOTUS, SYMPHONY, dBASE, PFS:FILE, REFLEX, VISICALC and ASCII. Note that when PARADOX exports a file to a spreadsheet format, it does not automatically generate a cross-tabulation of the data (see *Data entry*). When a crosstable is needed, the option CrossTab has to be se-

lected from the Image/Graph menu. For this option, the Quantification field has to be numerical. When the cross-tabulation has been performed, the results are placed in a temporary table called CROSSTAB. This table can be renamed and subsequently exported to LOTUS or QUATTRO PRO.

It is also possible to generate a presence-absence table, or some other transformation of data, which can then be used for statistical packages like SPSS or CANOCO.

Acknowledgements. We are very grateful to Nico Heerschap (Netherlands Central Office for Statistics; C.B.S.) for his valuable comments on the design of RADAR.

References

- Brandt RW, Drenth E, Montforts M, Proos RHP, Roorda IM, Wiemer R (1992) Archeologisch Basisregister, versie 1.0 [Archaeological database, version 1.0]. ARCHIS, Amersfoort
- Duuren L van (1993) Botanisch Basisregister 1993 [Botanical database 1993]. Centraal Bureau voor de Statistiek, Voorburg
- Greig JRA (1994) Pollen analyses of latrine fills from archaeological sites in Britain; results and future potential. AASP Contrib Ser 29: 101-114
- Meijden R van der (1990) Heukels' flora van Nederland [Heukels' flora of the Netherlands], 21st edn. Wolters-Noordhoff, Groningen
- Nieuwerkerken EJ van (1991) UTM grid: een voorschot op de toekomst [UTM grid: an advance into the future]. Nieuwsbrief European Invertebrate Survey - Nederland 20:9-14 (in Dutch with English abstract).
- Tomlinson P (1992) Design and implementation of a relational database for archaeobotanical records from Great Britain and Ireland. *Circaea* 10(1): 1-30