Relational database - structures in archaeology: ADS - an application in client-server-conception developed by means of CASE method

Ofen, Ulrich von

Historical Social Research, Vol. 18 — 1993 — No. 3, 22-34

Relational Database - Structures in Archaeology ADS - an Application in Client-Server-Conception Developed by Means of CASE*Method

Ulrich von Ofen*

Abstract: A summary of roman irontime settlements in northern Germany was published in 1985[1]. This publication seemed to give a good survey of number and names of places. It thus appeared to be a valuable source in the preparation of my diploma thesis about a special settlement of that time. For the district of Harburg (off Hamburg) Rotting just mentioned two places, but when finishing my preparations thirty-five places were known. Some of which had been already published. Consequently it is not only important to collect information, but also make it easily accessible. Therefore the idea was to create an information system for the archaeologist which can be used on an excavation site, the source of any archaeological information, as well as in a museum. Such a system should enable the user to structure and evaluate archaeological data of any kind. To provide such means an application called »ADS« (Archaeological Database System) on the bases of a Relational Database Management System in Client-Server-Conception.has been developed.

This abstract of a doktorate thesis is the attempt to transfer data from archaeological excavations into a useful Relational Database Management System (RDBMS)[2]. Main objective here is the technical development of ADS and not

* Address all communications to Ulrich von Ofen, Rosenstr. 2, D-45476 Mülheim an der Ruhr.

I would like to thank Prof. Dr. Bergmann for supervising the development, for introducing me to similar applications in social-history, and in general for his courage in managing this project. Furthermore thanks to Dr. Lorenz for his critical hints on the archaeological aspects. This work was made possible by Jauch & Huebener KGaA, an industrial broker in Mülheim, specially by Mr. Pehler, managing director, who gave me permission to use the Development-Tools listed below, which are neccessary to plan and to realize such an application.

[1] I. Rotting, Siedlungen und Gräberfelder der Römischen Kaiserzeit. Studien und Vorarbeiten zum Historischen Atlas Niedersachsens, Heft 31, 1985

[2] Date, C.J. (1983): »An Introduction to Database Systems. Volume II«, The System

the discussion of sense or nonsens of such systems in archaeology, which has not lead to new aspects so far. In todays world information is a key resource. Now the target is to ameliorate and to accelerate the flow of important information in archaeology, and thus support the scientist in answering their special questions. The information provided to scientist must of course be accurate, timely and complete. Keeping that in mind the idea was to develop an application running on a stand alone PC and also on a UNIX-Workstation under networkconditions in Client-Server-Conception. The most appropriate method to realize this enterprise is applying the »CASE*Method« (Computer Aided Systems Engineering). The resultant Database-application, is called »ADS« (Archaeological Database System).

2. The CASE*Method (overview)

The success of any systems design depends on achieving a clear base of understanding of the needs of an organization, and the environment in which it operates. In order to apply the CASE*Method it is important to look closer at the needs of archaeology and its form of organisation[3].

To develop an application from these needs the CASE*Method is divided into several stages (Fig.l). Each different stage or phase is much too complex to be described in detail here. Thus the main objective of this paper can only be to point out the most important steps in the development.

All the steps listed in Fig. 1 are part of the whole system-development most of which are described below. The strategy stage comprises the decision of what has to be developed and the collecting of relevant information. In the stage of analysis one determins how the enterprise is to be realised. One can reach this goal by defining functions, solving the problems of so called »unique identifiers«, and defining primary and foreign keys (explained below). In the Design-phase one transforms the gained results of the first two stages into a database - structure which contains tables and views followed by detailled descriptions of columns. The last step of pure software- or systemsengineering, is the implementation. This is the point of building the application and planning of the menuestructure and standard reports. Transition is the process of collecting technical advice and instructions to enable a correct configuration of the workstations within a client-server-architecture. The final phase is the produc-

---

Programming Series, Addison-Wesley Publishing Company, Reading, Mass., s.v. »Relational Database Management Systems« Alagic, S. (1986): »Relational Database Technology. Text and Monographs in Computer Science«, SpringerVerlag, Berlin 1986

[3] General aspects concerning the CASE* Method: Barker, R. (1990): »CASE* Method - Tasks and Deliverables«, Addison Weshley Publishing Company, Rea-ding, Mass. 1990

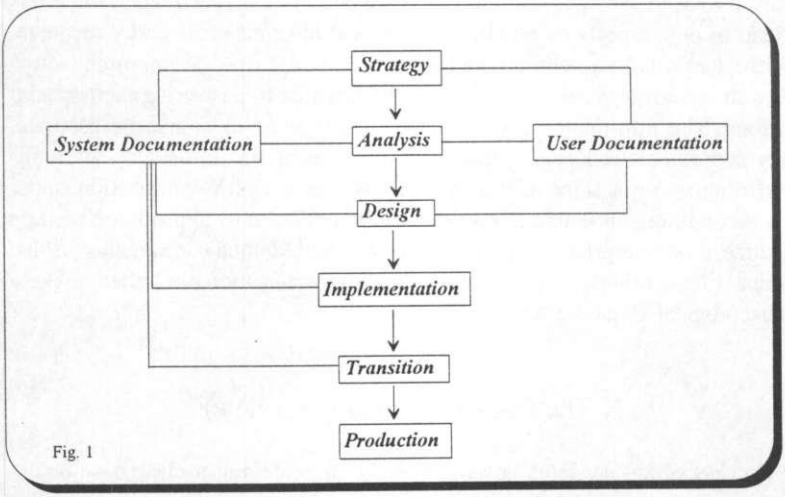


Fig. 1

tion, in which the ready to use application is set up. Besides production also means the maintenance of the program and to adjust to new requirements, just everything to guarantee a faultless operation. Of course all the above listed phases are components of a documentation. The user-documentation is a manual containing keyboard-functions, menue-structure and general advice concerning the realised application. The system-documentation in contrast is the inevitable background for the developer who maintains the application and who is responsible for support, including all technical aspects.

2.1. Strategy

The objective of the strategy stage is to produce an agreed plan for information systems development. To establish a good working relationship with all participants (the later Users) it is absolutely neccessary to create a common instrument of communication. This instrument is the »Entity Relationship Diagrams

2.1.1 Entity Relationship Modell (ERM)

An entity is an object of significance, real or imagined, about which information needs to be known[4]. At first glance this definition is not very useful. It becomes more tangible upon turning to the subject, Archaeology, and finding out the objects of significance (fig. 2). Each entity is framed and printed in bold

[4] Barker, R. (1990): »CASE*Method - Entity-Relationship Modelling«, Addison Wesley Publishing Company, P. GL-3

letters. An archaeological object of significance is for example the »Placer«. It can be defined by using some attributes like number, description, geological circumstances and etc. The procedure of defining these attributes within an entity is what the developer calls »Entity-Attribute-Definition«. After having defined all the objects of archaeological interest it is important to determine the relationships between them (relational database system !)[5]. The aim hereby is to incorporate both, user and developer. In order to achieve a common base of understanding one agreed on the following:

1. Lines represent relationships between entities.
2. Hatched lines are optional relationships, continues lines are mandatory relationships.
3. Splitted lines are used for 1 - n relationships, unsplitted lines are used for 1 - 1 relationships

Transferred into the ERM for ADS this means: One placer may have one or more findings, but if a finding exists that special finding must belong to that one and only one placer.

The pictorial representation of the Entity-Attribute-Definition and the Relationship-Definition is the Entity-Relationship-Modell, the first step of developing an application. At that point of time one already gets detailled information about the planned result of the application.

i.e.:

- Entering and reporting of findings and founds with a detailled description according to their place
- Entering and reporting of anthropological data, Dendro-data, data concerning mikro- or makrofragments, data concerning radiocarbonite - tests
- Entering and reporting Literature for each placer and
- Standard reports for getting statistics based on the information entered before, i.e.: katalogue of each place, combined statistics concerning pottery a.s.o.

Everything entered before based on the Entity-Relationship-Modell can be reported in any way.

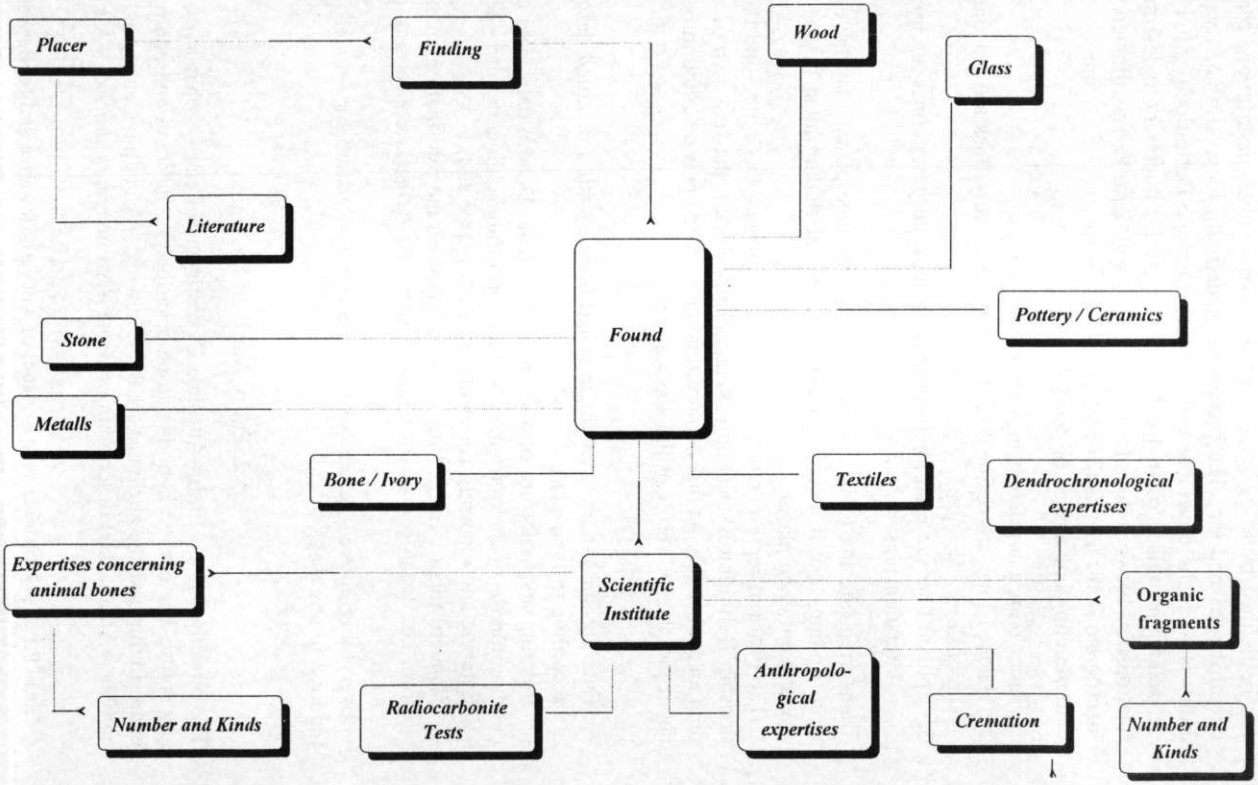
2.1.2. Entity Attribute Definition

The objective of the Entity-Attribute-Definition is to clarify what is meant by any entity. The entity »Found« can be described by defining and adding attributes. In this case the attributes are the following: +A unique identifying found-number +An attached description or name +An inventary number +A brief

---

[5] Chen, P.P.S. (1976): »The entity-relationship-model - toward a unified view of data«, in: ACM Trans.Database Syst 1, 1976, P. 9ff. Codd, E.F. (1970): »A relational model of data for large data bases«, in: Comm. ACM 13, 1970, P.377ff.

Fig.2 Archaelological Database System - Entity-Relationship-Modell



comment about its location within the finding +The number of the plate, by which the found will be illustrated +Dimensions (Length, width and height) +A short comment about the material the found was made of

These criteria sufficiently describe the entity found and the next entity can be described now. Very often founds are made of ceramics. It is usefull for such founds to create a seperate entity »Ceramics«, which can be described by the following attributes.

+Granulation +Figure of rim +Diameter of rim in cm +Largest diameter of the found in cm +Additives +Texture of outer and inner surface +Colour of the inner and outer surface and of the nucleus +Number of fragments present

The two entities described above are closely related. If pottery data has to be entered into the system, the detailed description consists of the attributes from the entity found and the attached attributes from the entity Pottery/Ceramics.

2.1.3. Function Hierarchy

The ERM in connection with the Entity-Attribute-Definition is consequently a summary of the data or information to be used within the planned application. The Function Hierarchy in contrast is a way to show what you are going to do with the information collected before. The handling of founds first of all requires the possibility of entering founds (fig.3). In some cases it is also useful to be able to delete data i.e. in case of wrong entering. Maintenance of data means to allow additional hints for updating, or to get more into detail after having posted it once. It is of course important to report a found, for statistical or other purposes. To set up queries or just to look at a found one has to be able to retrieve data. In short: five steps are neccessary to guarantee a comprehensive maintenance of a found:

- Entering
- Deletion
- Maintenance/Updating
- Reporting
- Retrieving

The above mentioned is only an example for the composition of a Function Hierarchy concerning founds. The compilation of all aspects (placer, finding, expertises a.s.o.) is the complete and detailled Function-Hierarchy for the planned application »ADS«. After having defined the Function-Hierarchy the Strategy-Phase in the development of »ADS« is completed.

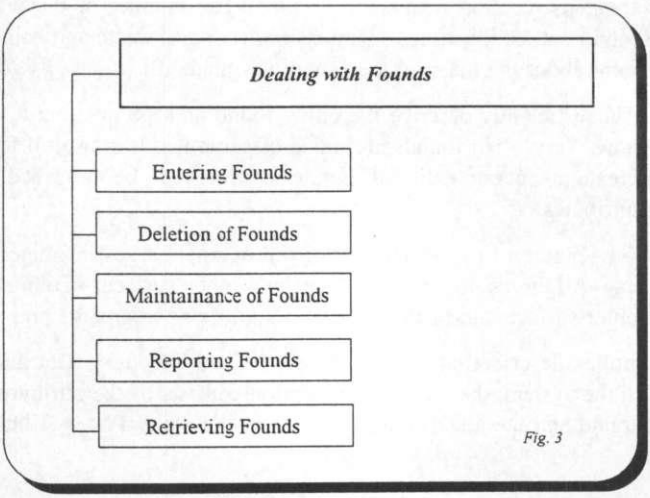


Fig. 3

2.2. Analysis

The analysis stage will take and verify the results from the strategy phase and expand these into sufficient detail to ensure accuracy and feasibility during the single steps of development[6]. In short: the Strategy phase is the answer to the question on what has to be developed, the analysis is the answer to the question how it can be accomplished. One part of the analysis is the Function Definition, which is the detailed description of the Function Hierarchy and the Unique Identifier Definition[7].

2.2.1. Function Definition

The objective of the function definition is the further sophistication of the results based on the Function Hierarchy. Here one combines different tasks within the function hierarchy with confering different rights in a technical sense (fig.4). Fig.4 represents a part of the function definition which is concerned with the entering of founds. In a first step one selects the entity »Found« in order to treat it the way determined by the function hierarchy. In the second step technical rights such as »create«, »update«, »retrieve«, »delete« and others are being placed. The confering of rights depends on the facts laid down in the

---

[6] Barker, Unit 4-3 Schmidt, J.W. / Brodie, M.: »Relational Database Systems - Analysis and Comparisons Springer, Berlin 1983

[7] It is important to remember that the single steps listed up within the several phases are only complete in view of ADS. Every developing of an application of cours depends on special requirements. Other requirements demand other steps during developing.

function hierarchy. It is only possible to place rights if a function hierarchy already exists. One part of the function hierarchy for example is the creation of a found. The task of the function definition is to express that fact by placing »Create«-Rights (fig.4) to the entity as such and to any correlated attribute. The deletion of a found is also a part of the function hierarchy. Within the phase elucidated in this chapter the hierarchy point »Deletion« will be defined by placing »Delete«-Rights again to the entity as such and to the correlated attributes. The function definition has to be prepared for all entities and attributes being part of the Entity-Relationship-Model and of the Attribute Definition for the whole application.

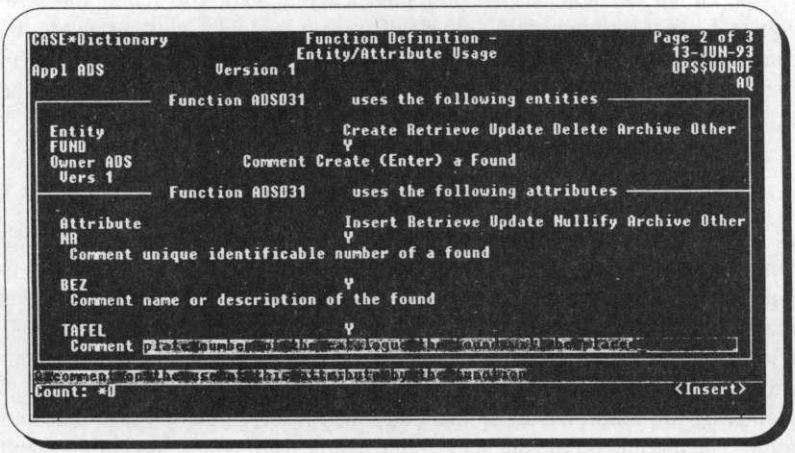


fig. 4

2.2.2 The Unique Identifier Definition

Within the Function Hierarchy and the Function Definition the main objective is to transfer the entities and attributes collected during the strategy phase into a programmed structure. The only remaining aspect is the problem of the unique identification of a found or any other entity which is inevitable to avoid problems concerning the relationship of a found to its finding or place. For example it has to be made sure that each finding only relates to one and only one found with the same number and description. In order to ensure that and to prevent the user from duplicate enterings which will falsify statistics and reports each unique identifier has to be defined within the system. These unique identifiers are the so called primary keys of the later tables originated from the entities. The place for example is uniquely identified by his number and description. This is also the case with a finding and the related founds.

After having defined the unique identifiers of all entities being part of the planned application, there is no chance for duplicate values in the system.

2.3. Design Phase

2.3.1. Default Database Design

The objective of the design phase is the transfer of the Attribute-Definition and the Relationship-Definition into a useful Database-structure. Now the entity can be regarded as the anticipation of the later table and the attributes as the related columns[8]. So far the table has not been physically created. The default database design is the first step in order to physically create the tables with respect to the remarks entered before. The next step, the so called column definition, is to revise the columns derived from the attributes of the former entities. Now the unique identifiers can be regarded as the primary keys of the tables.

2.3.2 Module Definition

It is important to have a thorough understanding of how an archaeological organization works in order to coordinate the building of Screens according to its special demands. This for example is the answer to the question, what is neccessary to enter a found ? Keeping in mind that before entering a found one has to select the attached place and finding one comes to the following conclusion: A screen containing blocks for the selection of placers and findings and entering the found has to be planned. Such a screen can be regarded as a module. The module on the one hand is the synthesis of tables and columns and on the other of the function hierarchy and function definition. The program consists of the modules generated for this application arranged within a menuestructure. Consequently a usefull archaeological database system like ADS has to be built up of modules guaranteing a faultless handling of placers, findings, founds, special materials, expertises, and Literature. The module definition however is not the only objective to plan and realize the later application the users will be working with.

A second part of the module definition is the determination of the requirered reports and statistics. After having completed the module definition with a description of the module detailled usage the application is ready to be transferred into a programmed structure. This has to be accomplished within the Transition phase, the last phase of pure software-engineering.

[8] General aspects concerning the Design-Stage: Chen, P.P. (1980): »Entity-Relationship Approach to System Analysis and Design«, North Holland, Amsterdam Gardarin, G. (1987): »Relational Database Systems. Design and Implementations Addison-Wesley Company, Reading, Massachussets 1987 Yao, S.B. (1985): »Principles of Database Design. Vol. 1: Logical Organisations^ Prentice Hall, Englewood Cliffs (NJ)

2.4. Transition

All the above listed steps are laid down and documented in a special tool, the so called CASE*-Dictionary. One part of this tool is the CASE*Generator that transforms all the made remarks into the needed programmed structure within the transition-phase. There are Generatorprocesses for module-definition, reports, statistics, and for additional helpfunctions. The 4th-Generation language, the CASE*Generator uses during these processes is SQL (Standard Query Language[9]). This language is used for the Creation of tables, the generation of modules, reports and statistics.

After having accomplished the CASE*Generator processes every single part of the program is ready to be used. The final step is the conception of a menuestructure, which transfers the different modules, reports and statistics into an application. Here you define main-menues, sub-menues and menues for getting reports and statistics. A sensible conception for archaeology could be the following which is already implemented in ADS. One part of the main-menue is »Placers«, the attached submenues are »Creation of Placers« and »Reports and Statistics«. The first point in the submenue automatically leads to the screen for the Creation of Placers. The second point in the submenue, the sector of the reports, again can be subdivided into three additional submenues,

- reports concerning districts in connection with dating and kind of place
- reports concerning districts in connection with soil and its actual cultivation or tillage
- reports concerning geographic hints (X- and Y - Coordinates)

The navigation scheme for the first point in the mainmenue as part of the systems-documentation, can be the following (fig.5). It is most appropriate to add the key-combinations for the elucidation of the navigation between the different blocks: The aim hereby is to summarize all features concerning places in a general mainmenue. After having defined and generated the navigation structure, the application ADS now is ready to use. It can be set up by an expert on the workstation or mainframe.This is not a standard software that should be installed without the aid of an expert.

3. Hard- and Software Requirements

As mentioned before, ADS can be setup as a stand alone Version on a lokal workstation or under networkconditions in a so called client-server-conception. The Hard- and Software requirements are listed below with respect to a stand alone Version. The client-server-conception will be dealt with in the next sec-

[9] Martin, J. (1985): »Fourth Generation Languages. Vol. 11«, s.v. »SQL«, Prentice Hall, Englewood Cliffs (N3)

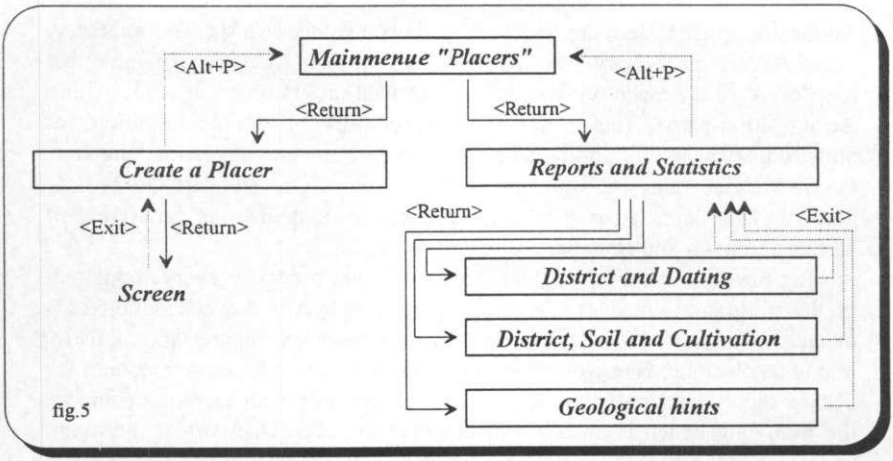


fig.5

tion. In order to comfortably run the stand alone Version of ADS the user should have the following Hardwarecomponents:

- IBM comp. PC 486/33Mhz, 16 MB RAM
- at least 60 MB free capacity on the local drive
- Laserprinter comp. to HP Laserjet IIIplus

Softwarerequirements:

DOS	Vers.6.0 in connection with WINDOWS, otherwise Vers.3.3 upward
ORACLE SQL	Version 6.034
ORACLE SQL-Menu	Version 5.0.11.9.4 (Building the menuestructure)
ORACLE SQL-Forms	Version 3.0.16.9.5 (Building Screens)
ORACLE SQL-Reportwriter	Version 1.1.13.2.3 (Getting reports)
ORACLE SQL-Net TCP/IP	Version 1.1.6.8
SQL-Plus	Version 3.0.10.1.4

ORACLE-RDBMS 6.0 and the runtime-modules of SQL-Forms, SQL-Menu and SQL-Reportwriter, the so called exécutables have to be installed. These exécutables take care of the correct processing for the whole application and are responsible for distribution of data on the tables within the RDBMS.

4. Client - Server - Conception

The Client-Server-Conception is part of a distributed database conception with the target of separating program and data. The program should run on the local drive and the data should ly on the server for an automatic Datastorage[10] and amelioration of the performance. The Application to be set up called »ADS« contains tables, views, ReportWriter- and Formsmodules. The exécutables and the PC - Software on the local workstation (Clients) should ly in different directories. In addition to these points the administratorś workstation has got the source-files and the SQL-Scripts for the Installation of Users (grants and synoyms) and the maintenance of tables (create-table-scipts or alter-table-scripts). There is the chance to keep the exécutables on the server in a shared directory. This means an increased workload for the network during the start of the application but in contrast it also saves a lot of time in terms of installations and modifications within the program. Having setup the database access the application is ready to use under client-server-conditions.

The database-objects (tables, synonyms, views, indices) have to be installed on the Server. Here you can also find the general access - rights of each user. Every user must have access to the server and to grants on database-objects.

5. Environment of Development

The development of ADS has been realized under the following Hardware- and Software conditions under UNIX:

Hardware:	
SERVER:	HP 9000 Serie 835 HP, operating system UNIX 8.0
LOCAL WORKSTATION:	PC HP-Vectra 486/25Mhz, operating system DOS 5.0
Software:	
ORACLE SQL	Version 6.034, SQL-NET TCP/IP[11]
ORACLE SQL-Menu	Version 5.0.11.9.4(Building the menuestructure)
ORACLE SQL-Forms	Version 3.0.16.9.5(Building Screens)
ORACLE SQL-Reportwriter	Version 1.1.13.2.3(Getting reports)
ORACLE SQL-Net TCP/IP	Version 1.1.6.8
SQL-Plus	Version 3.0.10.1.4

[10] Draffan, I/Poole, F.(1990): »Distributed Data Bases«, Cambridge University Press, Cambridge

[11] Martin J. (1985): Fourth Generation Languages Vol. II, s.v. »SQL«, Prentice Hall, Englewood Cliffs (N3) Date, C.J. (1987): »A Guide to the SQL Standards Addison-Wesley Publishing Company, Reading, Mass.

6. Literature

Alagic, S. (1986): Relational Database Technology. Text and Monographs in Computer Science, Springer Verlag, Berlin 1986

Barker, R. (1990): CASE*Method - Tasks and Deliverables, Addison-Wesley Publishing Company, Reading, Mass., 1990

Barker, R. (1990): CASE*Method - Entity Relationship Modelling, Addison-Wesley Publishing Company, Reading, Mass. 1990

Chen, P.P.S. (1980): Entity-Relationship Approach to System Analysis and Design, Noth Holland, Amsterdam 1980

Chen, P.P.S. (1976): The entity-relationship model - toward a unified view of data, in: ACM Trans. Database Syst. 1, 1976, P. 9ff.

Codd, E.F. (1990): The Relational Model for Database Management Version 2, Addison-Wesley Publishing Company, Reading, Mass. 1990

Codd, E.F. (1970): A relational model of data for large data bases, in: Comm. ACM 13, 1970, P. 377 ff.

Date, C.J. (1986): An Introduction to Database Systems, Vol. I, The System Programming Series, Addison-Wesley Publishing Company, Reading, Mass., 1986

Date, C.J. (1983): An Introduction to Database Systems, Vol. II, The System Programming Series, Addison-Wesley Publishing Company, Reading, Mass., 1986

Date, C.J. (1987): A Guide to the SQL-Standard, Addison-Wesley Publishing Company, Reading, Mass., 1987

Draffan, I.W. / Poole, F. (1980): Distributed Data Bases, Cambridge University Press, Cambridge 1980

Gardarin, G. (1987): Relational Database Systems. Design and Implementation, Addison-Wesley Publishing Company, Reading, Mass. 1987

Martin, J. (1985): Fourth Generation Languages, Vol I and II, Prentice Hall, Englewood Cliffs (N3) 1985

Schmidt, J.W. / Brodie, M.(1983): Relational Database Systems - Analysis and Comparison, Springer Verlag, Berlin 1982

Ullmann, J.D. (1982): Principles of Database Systems, Pitman Publishing Ltd., London 1982

Yao, S.B. (1985): Principles of Database Design. Vol 1: Logical Organisations, Prentice Hall, Englewood Cliffs (N3) 1985.