Archeologia e Calcolatori
7, 1996, 591-602

DYNAMIC CLASSIFICATION AND DESCRIPTION IN THE IDEA

1. Introduction

The IDEA (Integrated Database for Excavation Analysis) is a meta-database system for excavation recording and analysis being developed at the Moesgaard institute of Anthropology and Archaeology in Denmark (Andresen, Madsen nd). The system is implemented in Microsoft Access, and the advanced capabilities of this product is used to create a flexible and powerful tool for setting up databases for excavation recording.

Powerful in this context means that it is able to handle very complex recordings, and flexible means that it can be made to record information according to user specification of structure and conceptual content. There are two areas in particular where the IDEA lets the user customise the system. One is the data model of the excavations – the basic recording entities available, and how these entities are structured into a recording procedure for a particular excavation. The other area is the classification and description of deposits, finds and features.

Within both areas we experience severe problems with existing recording systems. Reasons given by archaeologists to reject a particular system is often either that the structure of recording is incompatible with the one preferred, or that the classification system and the description variables used are not those needed. For both issues solutions are provided in the IDEA. In this paper, however, we will concentrate on the classification and description problem.

2. The problem

Over the last thirty years it has been standard procedure in archaeology to set up formalised description systems for archaeological materials. The approach was developed in the fifties and sixties with Jean-Claude Gardin as the earliest and most notable proponent (Gardin 1958, 1967). Parallel to and nourishing this development was the advent of the computer. It was assumed that given a generalised, sufficiently detailed descriptive system, it would be possible to store an objective symbolic representation of archaeological materials on computers once and for all. Further, it was assumed that given this computer based objective description, it would be possible to implement statistically based automatic classification procedures (Borillo, Gardin 1974; Chenhall 1965, referenced in Scholz, Chenhall 1976; LeMaitre 1980; Voss 1967).

These initial optimistic ideas of global solutions of computer based storage and processing of extensive descriptive bodies of data soon died out (Audouze, Leroi-Gourhan 1981; Cleziou *et al.*1991; Hill, Evans 1972;

SCHOLZ, CHENHALL 1976). The task of setting up global descriptive systems was immense, but nothing compared to the workload of actually describing archaeological materials according to such descriptive systems:

> «The idea of the automated computerized information system is an image that many people hold, and one which our research has shown to be questionable. We have found the amount of clerical work required to produce the files described above to be tremendous, and there will be a continuing need for clerical work in the correcting and updating of the archaeological site file» (SCHOLZ, CHENHALL 1976, 94).

Worse, however, it was soon realised that researchers could use these immense descriptive bodies of data for limited purposes only, because they seldom met the specific needs of a particular project:

> If a data bank is to aid in scientific endeavours it must be planned and built to serve specific purposes. One soon realizes, however, that possible research designs are extremely numerous and diverse, and to attempt to record data for all possible analytical contingencies is impossible. Our research indicates that minimal units of observation are not inherent in the data, waiting to be discovered, and, once discovered, permuted to produce other analytic units or attributes. Minimal units of observation are problem specific, and they must be recorded as input to the data bank using terminology that will answer the problems under investigation (SCHOLZ, CHENHALL 1976, 94).

The point made here is that the inductive model of research, where an investigation is initiated with a thorough objective description of the material at hand followed by an analytical phase leading to some sort of conclusion is false. Instead Scholz and Chenhall clearly support the deductive model of research, where description is preceded by a problem formulation that decides what to describe. In terms of recording this means that each project needs its own data base designed after the problem formulation, but before the material is taken up for description. It is questionable, however, if even this model of research is correct.

READ (1990) has argued, that archaeological research can be viewed as a dialectic process between theoretical modelling on the one hand and data modelling on the other. This implies that both theoretical models and data models are dynamically changed throughout the research process. Thus, neither categorisation, nor description of data can be considered constant within a project. They can change continuously in response to insight gained. At some point, of course, a categorisation and description scheme has to be decided upon as final, but it need not be before late in the project, and following numerous changes.

The implementation of archaeological recording systems has traditionally been through the "coding sheet approach". In this the material is divided into description variables. Each variable has it own entry line on the sheet where a measure or one of a number of pre-defined nominal or ordinal attributes may be entered. When transferred to a database, each variable normally becomes a field. All variables may be placed in one table or in more tables linked in a hierarchical structure with one to many links. Very seldom the structure is truly relational, and whatever the classification structure behind the variables is, it is seldom reflected in the database.

In the inductive research model, where a description system is created once and for all, and even in the deductive model, where a description system is created at the beginning of the project, the hard-coding of the categories into the tables of the database causes little problem (well it does, but people are generally not willing to admit that their research model is wrong). If, however, we assume the dialectic model, and accept that alterations in the descriptive system can occur continuously, then we are in trouble, because not only do we have to adjust already made descriptions as we change the structure, but every time it happens we have to physically restructure the database tables, modify forms, etc.

3. The solution

Every time we adjust our description system we will have to adjust the descriptions of data already entered. No matter how the database is designed this will always be a problem. When we change the structure of the description system, however, changes of the physical structure of the database is not necessary if we create a meta-database structure within which any actual instance of that structure can be accommodated as a matter of user definition through the forms of the database. This is what we have tried to achieve with the IDEA.

Creating a meta-database is essentially a question of separating the conceptual content from the actual data records. In the tables holding the records (almost) only ID-numbers are stored. The conceptual content providing meaning for the data records is kept as data in another set of tables, where they can be entered into the database through forms like any other type of data. Actual meaning is assigned to the data records through cross references to the appropriate records in the tables containing the conceptual content.

An appropriate solution to the task of creating a meta-database for categorisation and description of data can only be achieved through a full relational DBMS (or object oriented DBMS). Hierarchical DBMS are not sufficiently advanced for the purpose. As mentioned in the introduction we have used Microsoft Access. This is from version 2 and onwards a full relational DBMS. In the following we shall try to demonstrate, how the solution is achieved, and for simplicity we will limit ourselves to the part of IDEA that deals with the categorisation and description of finds. Naturally, the parts that deals with deposits and features are structured in a similar way.

Record_ID	Class_ID	Object_ID	Variable_ID	Value
Record 1	Class A	Item 1	Variable 1	Value 11
Record 2	Class A	Item 1	Variable 2	Value 12
Record 3	Class A	Item 1	Variable 3	Value 13
Record 4	Class A	Item 2	Variable 1	Value 21
Record 5	Class A	Item 2	Variable 2	Value 22
Record 6	Class A	Item 2	Variable 3	Value 23
Record 7	Class A	Item 3	Variable 1	Value 31
Record 8	Class A	Item 3	Variable 2	Value 32
Record 9	Class A	Item 3	Variable 3	Value 33
Record 10	Class B	Item 1	Variable 1	Value 11
Record 11	Class B	Item 1	Variable 2	Value 12
Record 12	Class B	Item 2	Variable 1	Value 21
Record 13	Class B	Item 2	Variable 2	Value 22
Record 14	Class B	Item 3	Variable 1	Value 31
Record 15	Class B	Item 3	Variable 2	Value 32

Class A	Variable 1	Variable 2	Variable 3
Item 1	Value 11	Value 12	Value 13
Item 2	Value 21	Value 22	Value 23
Item 3	Value 13	Value 32	Value 33

Etc.

Class B	Variable 1	Variable 2
Item 1	Value 11	Value 12
Item 2	Value 21	Value 22
Item 3	Value 31	Value 32

Etc.

Fig. 1 – A meta-structure for object classification and description (left), capable of holding different class-determined description schemes (right). Arrows indicate how individual values of description tables are transformed into separate records in the meta-structure.

4. Conceptual and physical model for the meta-database

In accordance with the use in statistics, we prefer to term the minimal domain of description chosen in any investigation a variable. A variable is primarily characterised by possessing a name, a type and a set or range of values. That is: one variable has one name and one type, but several potential values. The set of values can be on different scales of measures (nominal, ordinal, interval/ratio). If it is on a nominal scale, multiple choice is a possibility, while interval/ratio scale requires a unit of measurement. With nominal and ordinal scales a set of alternative values must be provided. In case of an ordinal scale these values must be ranked as well.

Any variable has to be a variable of something. That is: some sort of classification has to precede the definition of the variables. You do not use the same variables to characterise a pot and a knife, so you have to categorise in advance what is a pot and what is a knife. Each pot then, will have a different set of variables than any knife. Basically, we have a situation, where a category of something is characterised by a number of variables, and where each variable in turn is characterised by a value drawn from a set of potential values.

We are used to record the description of a set of objects in terms of a table with the objects in the rows, and the variables in the columns. Consequently, objects of different classes and hence with different set of describing variables cannot be described in the same table. A basic demand of a meta structure is that any object, no matter what class and kind and what number of descriptive variables it posses, can be described in one table and one table only.

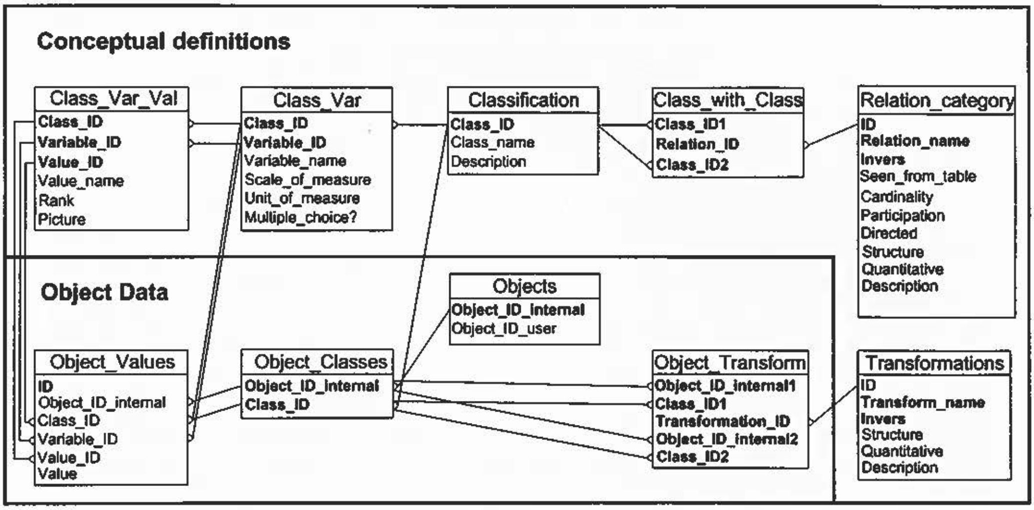

Fig. 2 – The IDEA physical design for holding the conceptual description and classification information of objects, and for holding information of actual object data.

Our solution to this problem is shown in Fig. 1. It simply takes every cell in the descriptive tables of the different classes, and transform them into a record of their own. This record holds all the information – Class ID, Object ID, Variable ID – of the Value entry. It may seem a very circumstantial approach, but then relational database methodology is very circumstantial, and although it may look beyond calculation, it is not.

It is of course possible to store the class names, item numbers, variable names and values directly in this structure, but it is not an acceptable solution. First of all it would be an enormous and wasteful work to type in all this information, as the same data are repeated over and over again in different combinations. Secondly, it would be an approach very prone to errors due to the risk of typing mistakes. Thirdly, it would be against relational theory, where one of the rules is that any piece of information should only be entered once, and subsequently referred to by way of cross references.

Thus, before we can start entering data into a table we need other tables in which to define the structure and content of the classification and description system. To set this up we need three tables linked to each other as one to many. That is: one class can take many variables, and one variable can take many values. In Fig. 2 we find this represented through the tables *Classification*, *Class_Var* and *Class_Var_Val*.

The actual data are stored in another set of tables. The *Objects* table holds the User_ID number as well as an internal number for the object, but nothing else. There is no actual information in the table, just identification numbers. Linked to *Objects* with a one to many connection is the table *Object_Classes*. This table contains only identification numbers as well, but one of these is a Class_ID linked to the ID of the *Classification* table, from where its conceptual content is taken. By placing the Class_ID in a separate table, and not in the *Objects* table we make it possible to associate more than one class with an object.

In a third table – *Object_Values* – a further two ID's are stored, and again the conceptual content referred to by these ID's are taken from the definition tables, either *Class_Var* or *Class_Var_Val* depending on what information is needed. In fact if we are dealing with nominal or ordinal variables there will not be one shred of descriptive data in the object data tables. Only for interval/ratio scale variables we store the actual value here.

By now, if not before, it should be obvious, why we can change content without changing physical structure. In contrast to traditional database design, all class entries relating to objects are kept in one field, all variable entries relating to classes are kept in one field, and all value entries relating to variables are kept in one field. The only "content" is the assertion that an object can be described in terms of a class, a variable and a value. This is the meta-structure. For each value, of each variable of each class of each object there will be a record in the database, and consequently for each object there may be many records, and not just one (Fig. 1). These records hardly consist of anything but ID's cross referenced to the content of the definition tables. Through these cross references meaning is assigned to the object entries.

A classification preceding a description need not be of a simple flat structure, and indeed a classification resulting from the analytical work very seldom has a flat structure. Some sort of hierarchical structure is mostly the rule, and the table *Class_with_Class* linked with the table *Classification* as many to many is set up to allow a hierarchical or even more complex structure to be described. In order to qualify the nature of the relationship between two classes an extra table *Relation_category* has been added. This contains the necessary definitions of the relationship to allow a correct handling in the forms and application programs and indeed to give the relationships a "meaning".

To round off this chapter it should be noted that the table *Object_Transform* linked to *Object_Classes* as many to many, makes it possible to describe if an object has undergone a sequence of changes of class (a flint dagger has been made into a strike-a-light, a fragment of a flint axe has been reshaped into a scraper, etc.). By adding a further table (*Transformations*) we have also made it possible to describe the nature of the transformation in detail.

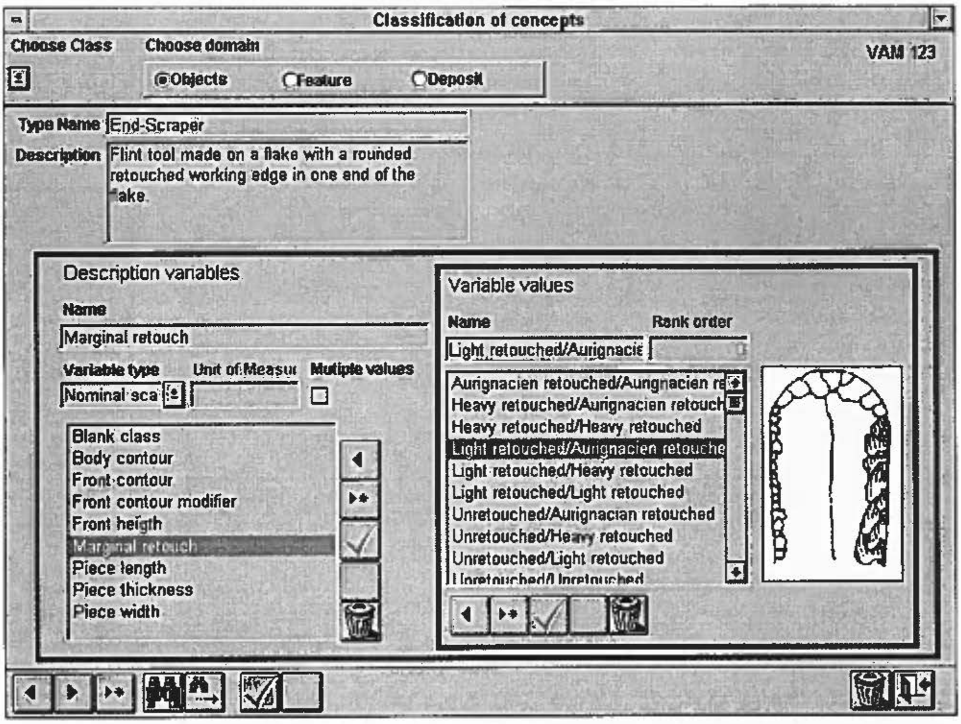

Fig. 3 – The IDEA entry form for setting up classification and description schemes.

5. Entering conceptual definitions

The basic set of conceptual information to be entered are as mentioned kept in three tables linked in a sequence with one to many relations (*Classification*, *Class_Var* and *Class_Var_Val* in that order). This offers a hierarchical structure as the maximum level of complexity, and setting up a form for entering data is thus fairly simple. For each table a form is created, and these are then embedded one within the other (Fig. 3). The outer form is directly associated with the *Classification* table. Classes can be entered, edited or deleted here, and only one class at a time will be displayed.

The next level form is associated with the *Class_Var* table. Through its embedding in the classification form its content is dynamically associated with the class currently displayed in the outer form. Thus it allows the entering, editing and deletion of variables associated with the current class, and it will simultaneously display all variables associated with this, although of course, only one variable is current and editable at a time. Apart from entering the variable name, the type, unit of measurement (for ratio/interval scale data), and allowance for multiple values (for nominal scale data) may be entered.

The third level form is associated with the *Class_Var_Val* table. It is embedded in the Class_Var form, and its content is dynamically associated with the variable currently displayed in that form. Thus it allows the entering, editing and deletion of values associated with the current variable, and it will simultaneously display all variables associated with this. The form will only be active if the variable type is either nominal or ordinal, though. In both cases it is possible to enter value names, and if the variable type is ordinal, it is also possible to enter its rank order. Further, there is a field in which pictures may be entered. For each nominal or ordinal value a picture may be attached. The field is an OLE control (Object Linking and Embedding), and the user may thus enter and edit the pictures through a graphical editor of own choice.

6. Entering object data

Through the *Artefact entry* form in IDEA we may enter the relations between objects and other information from the excavation, internal relations between objects (refitting, etc.), and the classification of the objects. Only the latter issue will concern us here. In Fig. 4 we see the top of the *Artefact Entry* form in the background. We start by entering the identification of the objects number – here "artefact x". Next, we enter the type or class of the object – here "End-scraper". For each object we may enter as many classes as we wish, as in principle there is no limit to how many classifications that may apply to an object. We have not yet implemented the use of the *Object Transform* table in which we can track the structure of complex class assignments.

The next step in data entry is to set the values for the descriptive variables. This is achieved through a pop-up form activated through double clicking the class selection field. This form contains three list boxes. The left list box shows all the descriptive variables recorded for the particular class End-scraper. When you activate a variable in the left list box, the middle list box immediately shows the values available for this variable (nominal and ordinal scale only).

Attached below the list box is a graphical control displaying the pictures associated with the values. Whenever a value is activated in the listbox the picture attached to this value will be shown. Double clicking of a value in the middle list box will record this value with the variable, and the recorded variable-value combination will appear in the right list box. The latter is continuously updated, and shows the current recorded values of the individual variables. When a ratio scale variable is activated in the left list box the middle list box will disappear and in stead a normal entry field will appear in which you can enter a number.

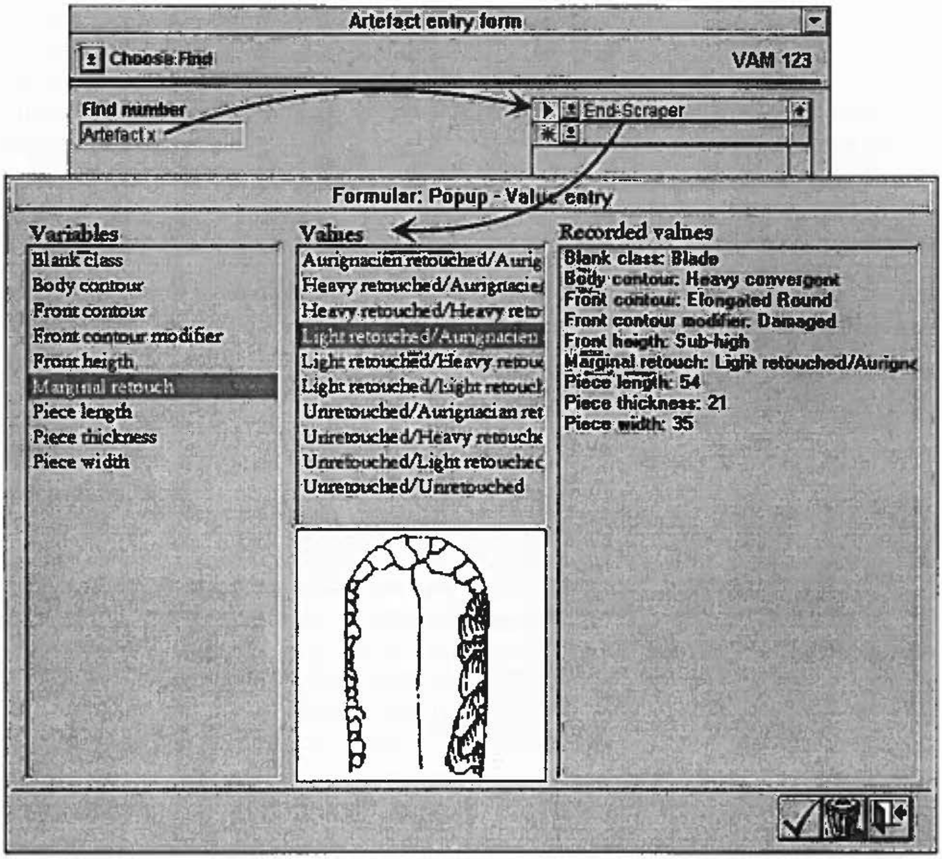

Fig. 4 – The IDEA entry forms for setting the class(es) of an object (at the back), and describing the object according to the chosen class (in front).

7. Retrieving object data

It is of no value to be able to store classifications and descriptions in a complex structure of tables, if you cannot retrieve the data again in a way that is intelligible to archaeologists. That is, we have to be able to extract the information in the tables class by class, with the objects in the rows, and the variables in the columns – the way we are used to see them. At the moment we can do this through a considerable programming effort, where from code we dynamically create a temporary table with the needed number of columns and correct headings of these for the class in question, and then, still from code, fill in the table as we read though the recorded data.

However, before we even began to consider implementing such a solution, we became aware of a Visual Basic custom control called Grid. This control will do exactly what we need with hardly any coding necessary. It will be able to present the descriptive data on screen in table format as a response to a query on any object class (number of rows and columns are dynamically set), and further the grid data can be easily transferred to other programs for say statistical treatment. One limitation is rather worrying, though. The control cannot have more than 2000 rows, which is obviously a too limited number in a real life situation. For the time being it will do, however, but later on we will have to decide whether it is necessary to go into a work-around for this problem.

8. Conclusion

The IDEA is a system aimed at recording and analysis of excavation information in general. Its central issue is to integrate various information entities from excavations into a flexible structure that may be customised by the end user (Andresen, Madsen 1992). Finds constitute one such basic entity, and the classification and description facilities described here is an extension to be used in connection with "specialist studies" of the finds material.

In this context the advantage of the meta-structure described is obvious. When you design the recording system for a particular excavation to be carried out, it is impossible to know exactly what find material you will come across. You may not even know the specialists to become involved at a later stage of the project, and you will certainly not know their requirements with respect to classes and descriptive variables. As a result, it is seldom, if ever, seen that the recordings of the specialists are integrated with the excavation recordings. Specialist databases live a life of their own in the custody of the specialists, and may never become part of the excavation documentation kept by the responsible institution. The IDEA effectively solves this problem.

As discussed previously in this paper the understanding of the archaeological research process has changed substantially over the years, and parallel to this our attitude towards description and classification has changed. The notion of fixed, pre-set description systems has been replaced by a more dialectic attitude, where alterations in the descriptive system can occur continuously. Clearly, the IDEA meets these requirements.

The actual implementation of the meta-structure presented here is fairly simple, but the *Class_with_Class* table holds potentials for further development. It allows us to define complex classifications, and we may use its information to try to implement inheritance through the class structure. That is: variables defined on a higher level of a hierarchy could automatically be inherited by lower level classes, and thus only be defined once. This would make it easier to create and modify a description structure, and it would secure consistency.

Another potential that we should look into is a shift away from hierarchical decomposition as a basis for description systems and towards relational description. As demonstrated by DALLAS (1992), relational description has definite advantages compared to the traditional fixed hierarchies. It will not be easy to control, but it would be worth while if a methodology of relational description could be implemented. The problems we can anticipate will not be with the table structures nor with the entering of data into these, but definitely with the methodology of extracting and utilising the information in a proper way.

JENS ANDRESEN, TORSTEN MADSEN
Moesgaard Institute of Anthropology and Archaeology
University of Aarhus

BIBLIOGRAPHY

ANDRESEN J., MADSEN T. 1992, *Data structures for excavation recording. A case of complex information management*, in C.U. LARSEN (ed.), *Sites and Monuments. National Archaeological Records*, København, Nationalmuseet, 49-67.

ANDRESEN J., MADSEN T. in press, *IDEA – the Integrated Database for Excavation Analysis*, (paper presented at the CAA 95 conference in Leiden, March 1995).

AUDOUZE F., LEROI-GOURHAN A. 1981, *France: a continental insularity*, «World Archaeology», 13, 170-189.

BORILLO M., GARDIN J.-C. 1974, *Banques de Données Archéologiques*, Paris, CNRS.

CHENHALL R.G. 1965, *An investigation of taxonomic systems for the storage and retrieval of material culture data on computers*, Unpublished Masters thesis, Arizona State University, Department of Anthropology.

CLEZIOU S. *et al.* 1991, *The use of theory in French archaeology*, in I. HODDER (ed.), *Archaeological Theory in Europe. The Last Three Decades*, London, Routledge.

DALLAS C. 1992, *Relational description, similarity and classification of complex archaeological entities*, in G. LOCK, J. MOFFETT (eds.), CAA91. *Computer Applications and Quantitative Methods in Archaeology 1991*, Oxford, BAR International Series S577, 167-178.

GARDIN J.-C. 1958, *Four codes for the description of artifacts: An essay in archaeological technique and theory*, «American Anthropologist», 50, 335-57.

GARDIN J.-C. 1967, *Methods for the descriptive analysis of Archaeological Material*, «American Antiquity», 32, 1, 13-30.

HILL J.N., EVANS R.K. 1972, *A model for classification and typology*, in D. L. CLARKE (ed.), *Models in Archaeology*, London, Methuen & Co, 231-274.

LEMAITRE J. *et al.* 1980, *Le RIDA. Réseau d'information et de documentation archéologiques: essai de réalisation (1974-1979)*, Paris.

READ D.W. 1990, *The utility of mathematical constructs in building archaeological theory*, in A. VOORRIPS (ed.), *Mathematics and Information Science in Archaeology: A Flexible Framework*, Studies in Modern Archaeology 3, Bonn, Holos Verlag, 29-60.

SCHOLTZ S., CHENHALL R.G. 1976, *Archaeological data banks in theory and practice*, «American Antiquity», 41, 1, 89-96.

VOSS O. 1967, *Dokumentationsproblemer indenfor arkæologien* (extensive English summary), «Kuml. Årbog for Jysk Arkæologisk Selskab», 1966, 97-134.

ABSTRACT

In the fifties and sixties it was assumed that a generalised and detailed descriptive system for archaeological materials could be constructed, and that this system could be transferred to a computerised symbolic representation. In the early seventies this position was abandoned as it was realized that data are theory-dependent and problem-specific. As a consequence it has been widely accepted that databases containing archaeological data are bound to be highly individual and short-lived. With the increasing number of IT-based archaeological recording system, the inherent heterogeneity becomes a hindrance for archival purposes and effective management of archeological projects. It is also, however, an obstacle for a formalized methodology, because researchers end up with pragmatic ad hoc solutions, which often shoehorn the recordings into rigid data-structures. The IDEA (the Integrated Database for Excavation Analysis) is to solve this problem. Through a database meta-structure and a user-friendly interface the IDEA offers the researcher the possibility to implement a problem-specific description of archaeological objects, but at the same time stores data and data-definitions in one underlying structure, regardless of chosen solution. In this paper we describe how we have solved the problem of creating a database structure capable of holding widely diffusing classifications and descriptions.