

## 4. Programming an intelligent database in hypertext

Juan A. Barceló

*Dept. d'Història de les Societats Precapitalistes, i d' Antropologia Social, Facultat de Lletres. Edifici B, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain.*

### 4.1 What is an 'intelligent' database?

We live in a world full of all kinds of information; however, most of the time we are not able to use it. Even as scientists we are not always able to manage all scientific information; it is far too great and diverse. The only possible way to resolve this dilemma of information management is by means of computers.

A scientist needs *structured knowledge* to solve scientific problems and not single amounts (even if they are very great) of empirical data. Therefore, we must build a kind of 'Intelligent' Database with more developed knowledge representation techniques than the classical relational databases. Such Intelligent databases will allow automatic inference, meaning the combination of data and rules, which will in turn produce new data. The aim of that System would be to solve scientific problems using a greater quantity of knowledge available in a specific scientific domain.

In an Intelligent Database the main aspect is not the physical implementation of individual data but rather the specific use of the knowledge, which it already contains. We must not only be able to retrieve a document, but we must also be able to use this document in a problem solving procedure. Consequently, according to the General Problem Solving Theory in Artificial Intelligence (Newell & Simon 1972, 1976; Newell 1973, 1980; Simon 1973, 1979, 1983; Pearl 1985; Laurière 1986; Brown & Chandrasekaran 1989) an Intelligent Database is nothing more than a *problem space*, which means that each knowledge unit in an Intelligent Database is one of the many possible states (solutions) of a problem or a set of homogeneous problems.

The main characteristic of an Intelligent Database is its internal architecture, which simulates scientific reasoning (in a particular domain). It automatically uses the same information in a similar way that a scientist would in doing research. Therefore, in order to build this sort of Database it is necessary to have an explanatory process scheme in scientific research to be translated into computational terms (see Langley *et al.* 1987; Churchland 1989; Thagard 1988; Shrager & Langley 1990). If scientific research may be divided into *reasoning about data* and *reasoning about hypotheses*, an Intelligent Database will contain two components: a mechanism to handle hypotheses (a rule base) and the proper intelligent database into which we have introduced validated hypotheses, theoretical principles and general information that we consider 'valid' in a specific domain. While the rule base is an Expert System with a somewhat classical architecture, the Intelligent Database component is very different from an expert system fact base. In this paper I will only deal with the second component; there is an abundance of literature already available on the first one.

Besides, we must consider the intelligent database component as the computer representation of a particular scientific theory. Then, we must program an intelligent database by describing and by representing the dynamic knowledge units it contains. As a result the database becomes more 'intelligent', so that to continue increasing this 'intelligence' we must incorporate:

- a set of concepts describing some relevant knowledge related to a problem and its meaning;
- an adequate 'active' representation of these concepts and meanings to allow the reaction of concepts to messages sent by the user or by other components of the system;
- a set of rules which will manage the concept descriptions in terms of their representation;
- a set of operators for the representational language;
- three kinds of meta-knowledge (Pitrat 1990):
  - knowledge on the problem to solve,
  - knowledge on the structure of the system,
  - knowledge on the strategy to solve the problem.

We call all *concepts* 'explanatory' elements in the database. These units are very different from empirical data because they contain 'knowledge', that is to say, 'a justified true belief', (Waern 1989). Concepts in an intelligent database are knowledge states, that is to say, the information about a problem which is available to a problem solver at a given moment in problem solving. Empirical data are the *result* of a series of *observations* (Bunge 1983). Without knowledge of the aim of the experiment, experimental results do not produce information. Therefore, if we must use these data, we need *knowledge* (justified true beliefs) about the precise way in which to use them to solve a particular scientific problem.

An intelligent database contains both declarative and procedural knowledge:

- declarative knowledge is a knowledge about facts or information about a particular scientific domain;
- procedural knowledge is knowledge about the way of using declarative knowledge, in other words, the way in which to solve problems in a particular scientific domain (meta-knowledge).

Both kinds of knowledge are integrated in the same unit, and as a result, meta-knowledge appears linked to factual knowledge. This is very important because it allows us to define 'knowledge-based' control structures as an alternative to the heuristic. If we want to use a

database to solve a great diversity of problems, we do not have to declare all possible problems and solving strategies. It is possible to use a set of the most abstract concepts in the concept hierarchy as the set of particular *goals* that the database can answer effectively. Then, solving heuristics would not be external — user defined — but internal, particular to this database and derived from the particular knowledge it contains.

Given the fact that an intelligent database is a kind of multi-expert system, it would be favourable to use an expert system *shell* as a programming language to implement the intelligent database. It is not a single software bridge between a classical expert system and a relational database (see Kerry 1990), but the integration of both modules. We have to use the advanced knowledge representation techniques typical of second-generation expert systems (like Nexpert/object, ART, KEE, IntelligenceCompiler, etc.) to program the semantic content in databases.

By using these kinds of representational languages, we obtain a *concept hierarchy*, more similar to an object-oriented architecture than to a taxonomic tree. It is not a 'heuristic' hierarchy (Clancey 1985) used as a search procedure, but a 'cognitive' relationship linking some concepts in terms of their knowledge contents. In a scientific theory there are many different hierarchical relationships among theoretical entities, however at this time I will only be dealing with differences in generality and observation content: concepts at the bottom of each hierarchy chain are those defined in observational terms (e.g. Weapon, Similarity, etc.), concepts at the top of each hierarchy chain are theoretical entities defined exclusively in theoretical terms (e.g. State, Social Stratification). It is the same ordination of hypotheses in a scientific theory. There are hypotheses with non-observational predicates, and they cannot be reduced to a conjunction of empirical data. The aim of all scientific research is to find the better connections between hypotheses at different observational levels (Bunge 1983; Laymon 1984; Hooker 1987).

Hierarchy only affects homogeneous units, which means that not all concepts are linked to the same hierarchy. Knowledge units are organized into *classes* and *objects* (Cox 1987; Tello 1989; Ferber 1990): objects are the minimal meaningful units (individual concepts), and classes are a set of hierarchically linked objects. In a database there are a great number of classes whose links are not explicit but hypothetical. Hierarchies within a class may be considered 'natural' relationships between similar objects. All other relationships between concepts (particularly relationships between classes) always constitute a new hypothesis, not included in the database component but in the rule base before its validation.

The system works in the following way. First, a global variable is created by the system whose content is an initial state for the problem we want to solve (this initial state is a description in observational terms). At this stage, the user suggests the goal he would like to investigate, but it is the system which will produce the best initial state for such a problem according to the

recognition of the goal as one of its higher-level theoretical entities. Because all concepts are linked, the system will find the low-level entities most adequate to the goal formulated by the user.

The following steps in the problem space are not guided by user goals, but by the individual concepts. They are active computational units capable of reading the actual state of the global variable. They are also capable of introducing some modifications according to the particular declarative knowledge the concept contains. Each concept works as a specialized expert system, which is linked indirectly to others. The spreading activation of the concepts depends on the actual state of the global variable and on the characteristics of each specialized expert system that implements each concept. When the global variable activates the high-level entity, which the system has recognized previously as the goal, the problem solving procedure reaches the solution.

## 4.2 Hyperdocuments and intelligent databases

In the case study reported in this paper I have used a hyperdocument generator and not an expert systems shell. The reasons being, first, because a hyperdocument is more transparent than an expert system, and second because we can effectively implement intelligent databases in hyperdocument format and represent knowledge in natural language terms. My goal is to explain the general architecture of intelligent databases rather than to build a real example.

'Hypertext systems contain frames of text, pictures, sound and animation that are organized nonlinear, in a network of linked frames. From any particular frame, users can access a variety of other frames containing text or other media ... Browsing is the typical means of accessing information in multimedia documents for both readers and authors. Users follow various sequences of frames and links to retrieve the information they require, or to add new frames and connections between them', (Foss 1988: 83; see also: Balpe 1990; Nielsen 1990; McKnight *et al.* 1991).

A hyperdocument is an Information System, which allows access to great quantities of information and which shows only the relevant part to the user's requirements. It is a computer system capable of reducing the complexity of a problem space (the set of all information units in the hyperdocument, usually texts and pictures). This reduction does not depend on some general instruction, the same for all units, but on the specific interaction with the user, who always selects the linking between different units. In fact an important part of the information produced by the system is produced by the users themselves who choose their directions.

One may ask: "is a hyperdocument a kind of intelligent database?" We would then answer, "in some cases this is true." If we were to build an Intelligent database as a set of individual texts, each one providing the declarative knowledge core of a concept, and if we

were then to implement the links between these texts in such a way that they were to work as a representation of the procedural knowledge, then we would obtain an intelligent database written in hyperdocument format.

Both intelligent databases and hyperdocuments lack a regular structure to represent their knowledge units. These units are totally independent. Some concepts have much declarative knowledge, but with a very simple attached procedure. Other concepts are very easy, weak in declarative knowledge, others are very complex, and their use is very difficult. There is no rule to define concepts because knowledge units are single entities; this is one of the main characteristics of intelligent databases and hyperdocuments, and one of the main differences with relational databases. The consequence of this is the dispersion of information, even though this may not pose too much of a problem because each unit has its own procedural knowledge content.

The objective of hyperdocuments and intelligent databases is not to retrieve a particular information unit as a result of some query, but to navigate (*browsing*) between information units and to extract information from this navigation: links between concepts (or texts) produce knowledge. In hyperdocuments, this knowledge is used freely by the user, and it changes according to the user preferences. In intelligent databases this knowledge is used by the system to solve a problem formulated by the user, and it is independent of user preferences. In hyperdocuments, control structures are guided by the user's preferences, while in intelligent databases, control structures depend on the particular knowledge that each unit contains.

Although the organization of information is approximately the same, there are differences between the kinds of procedural knowledge needed to browse through the systems. Both hyperdocuments and intelligent databases are represented as networks of nodes and links. The intelligent database and the hyperdocument have the same nodes; nevertheless, intelligent databases need a kind of 'intelligent' link that does not exist in most hyperdocument generators (e.g. in Guide or Hyperties). These systems are not conscious of the information they contain because the objective is to permit users to find information based on their own specifications.

If browsing depends on prior knowledge, then links must therefore be more complex. A solution would be to anchor a specialized Expert System in each link. This is possible in many systems (Nexpert/Object, Mahagonny, MacSmart, cf. Bielawski & Lewand 1991), using HyperCard as hyperdocument generator. This expert link has to be able to read the declarative content of the source concept and to introduce some changes in the destination concept. I have used this strategy to implement ESTELAS.

#### 4.3 A case study

The description of the ESTELAS intelligent database that I am proposing is not an operative implementation of the system, but rather a demonstration prototype to show some of its capabilities.

Materials used come from an analysis of a series of Iberian peninsula Warrior Decorated Stelae. They constitute the only archaeological record available to study social organization during the Late Bronze Age (11th–8th centuries B.C.) in the southwestern Iberian peninsula. The core of the intelligent database is a scientific theory on the origins of social stratification and the prehistoric state (Barceló 1989, in press); it is important to add, however, that the computer representation of this theory is only demonstrative, an actual implementation is already underway.

The reduced version of this theory suggests: 'sociotechnical items are used as symbols for social identity, and they act as one of the causes for social evolution.' In a specific region in the southwestern Iberian peninsula (the middle valleys of the Tajo and Guadiana Rivers), the social division of work reinforces the differentiated identity of specific prestigious social groups (people represented in the stelae). The progressive social independence of these groups ends in the effective control of the community exchange channels and the development of social stratification.

This process is characterized by transformations in material definitions of prestigious identity. Symbols are the same (swords, shield, helmet, etc.) but their social uses have become progressively different. In a specific phase, the ownership of a single identity symbol is substituted by the accumulation of sociotechnical items. Obviously, something has changed in this society. Traditional symbols of power and prestige are no longer effective because rivalry and social tension have substituted previous social balance and there are no new symbols to be used in this new situation. Therefore, the accumulation of prestige items can be considered as a new way to define social identity. In studying the modalities of social accumulation we will be able to discover the direction of social evolution. The analysis of frequencies and similarity relationships between the symbols represented in the stelae will produce a study of social accumulation in that age.

#### 4.4 Overview of this intelligent database

The description of stelae constitutes the initial state of the problem. The goal, which we are interested in, is the nature of social order associated with some particular disposition of symbols of power in stelae. That is to say:

- Which are the social correlates of Iconographic Homogeneity/Heterogeneity in a particular set of stelae?
- Which are the social correlates of the kind of social accumulation we are observing in this particular set of stelae?

One must assume that stelae represent an example of social accumulation before a fully stratified society. It is important to take into account that the goal is independent from the initial state; the same problem may have different initial states. This property will be used later.

First the system assumes that the user has asked for a specific problem (this operation is not implemented in the actual version). The user introduces some descriptive features (number of elements, iconographic complexity, etc.) which are read by the system and stored in a working memory. This working memory begins to 'travel' through the concept hierarchy, activating some concepts and receiving modifications. The last of these modifications will be the requested solution.

The procedure may be divided into the following operations:

- *Activation* — the user introduces some empirical data representing a real phenomenon (description).
- *Creating an Initial State* — a representation of the problem initial state is created automatically using heuristics derived from the problem and not from the knowledge. In the actual version, the system chooses some of the descriptive features introduced by the user: the presence of many/few symbols, individual variability/group diversity or iconographic homogeneity/heterogeneity among the stelae with the same geographic origin and chronology. Other descriptive features are ignored at this stage.
- *Calling Knowledge Units* — A previously created initial state works as an activation unit for knowledge available in this domain. The activation of theoretical entities depends exclusively on the successive states of working memory, as interpreted by a series of specialized knowledge-based rules built on each unit.
- *Using the Concept Hierarchy* — There are two kinds of links between knowledge units: hierarchy relationships (directed and explicit links) between the objects within a class and causal hypotheses ('intelligent' links) between classes.
- *Transforming Previous State* — each knowledge unit 'sends' some piece of knowledge to the working memory. According to specific instructions built in each concept, this new knowledge is added or changes the previous state of the problem. Of course, initial state (empirical data) will not be modifiable.
- *Constructing a Final Solution* — The answer is not the last activated theoretical entity, but the last state of the working memory once the last concept has been activated.
- *Reactivation* — Once we have a solution, then we need a validation. To do this, we must descend the hierarchy chains that we have used and explore alternative forks. For example, if iconographic heterogeneity in a geographical and chronological series of stelae has allowed us to confirm the existence of an important social differentiation in this community, then in the reactivation mode the system chooses an alternative to the relationship 'iconographic heterogeneity → social

differentiation', perhaps 'functional diversity in settlement structures → social differentiation'. In other words, in order for the system to confirm the existence of social differentiation in this community, we must to discover if there is iconographic heterogeneity and functional diversity. Keeping in mind that the same problem has different initial states (ESTELAS uses alternative initial states to validate its results).

#### 4.5 A hyperdocument implementation

The actual version of ESTELAS is written in HyperCard. Knowledge units are implemented as cards with text fields in which a declarative knowledge core has been written. Different backgrounds and visual effects allow us to distinguish different reasoning process and information categories (Data vs. Knowledge). A summary of all the links between concepts appears in Fig. 4.1.

This hyperdocument begins by asking for the name of a stele. After spelling correctly the name of one of the 57 known stelae, the system displays a drawing and some geographical information of the particular case requested. At this stage, the user can select another stele or can ask for the chronology.

The next linked card shows the chronology of this exemplar and a set of descriptive features characteristic of all contemporary stelae in the same area. The user may browse to other cards, which explain why the system has selected such features or uses another chronological hypothesis.

'PROBLEM ANALYSIS' is the first 'intelligent' link. The function of this procedural knowledge unit is to create the problem initial state, defining some global variables (working memory containers) and introducing the descriptive features that will activate successive knowledge units. In the figure I have represented those global variables as the first three items in a bold rectangle ('Iconographic Homogeneity', 'Iconographic Heterogeneity', 'Social Accumulation').

Global variables are invisible to the user, but their contents change after the activation of the successive intermediate states. After the activation of the 'PROBLEM ANALYSIS' link, for example, a new card is activated, and the working memory (in the beginning, empty) receives some initial information: the existence of iconographic homogeneity, iconographic heterogeneity and/or social accumulation in a subset of stelae. Always after the activation of a new card (or successive intermediate state of the problem) the working memory acquires new information, depending on the specific content in this card. Global variables remain invisible until the last card in the hyperdocument.

The unit 'Sociotechnical Item' is the first intermediate concept activated by the system after the definition of the initial state (the activation of the unit 'Subsistence' would reveal a contradiction in the construction of the initial state). It contains a direct link to the unit 'Kind of Accumulation', which contains a single conditional fork: the 'Controlled Distribution' or 'Uncontrolled

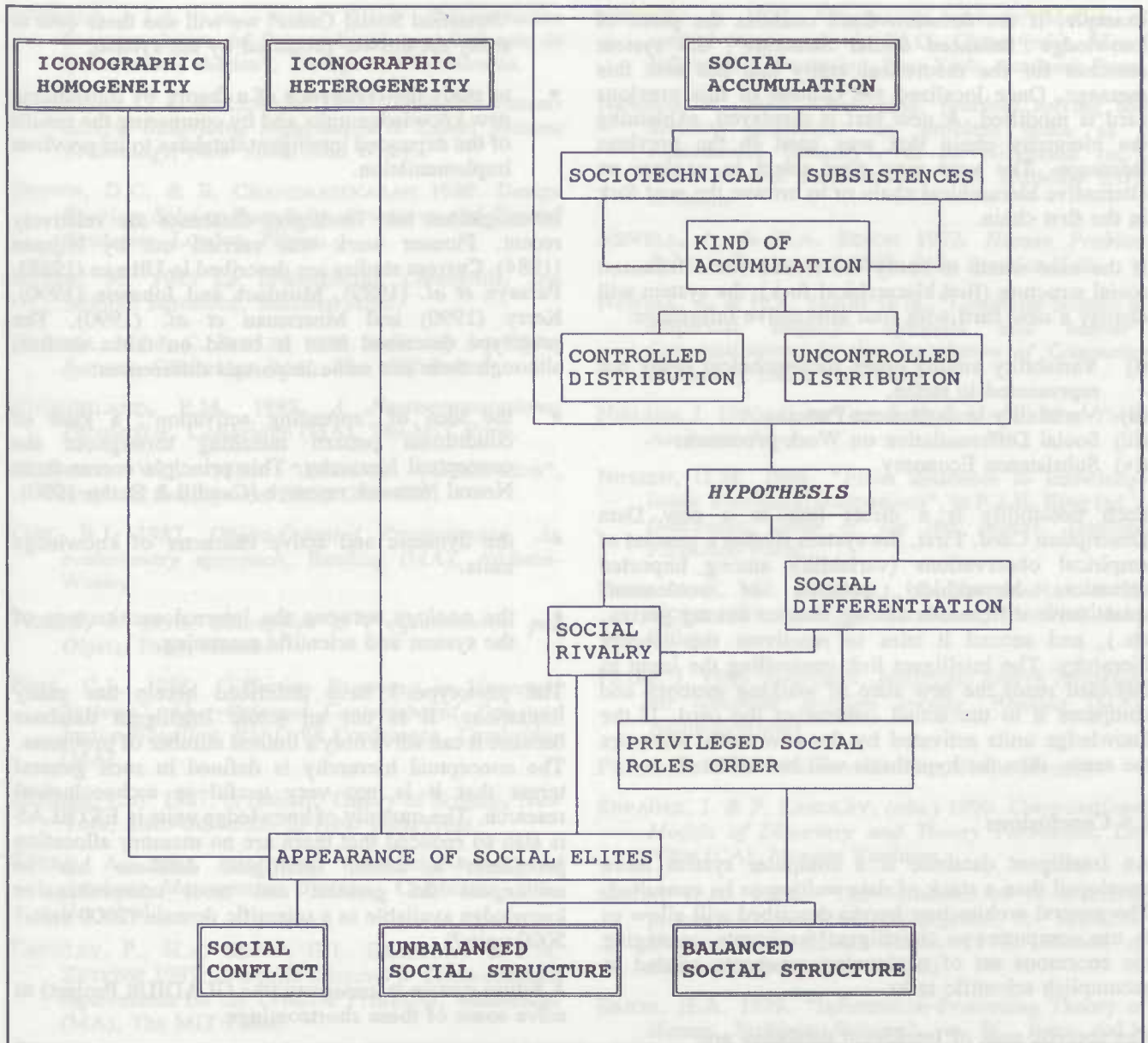


Figure 4.1: Order of activation and inner structure of the ESTELAS Conceptual Network

Distribution' units cannot be active for a single data set. This unit contains a procedure capable of using the information in the working memory to select the most appropriated. This procedure is an Expert System whose factbase is constituted by the actual state of the working memory.

All these units or cards constitute a single class of concepts related by means of direct links. The next 'intelligent' link (labelled 'hypothesis' in Fig. 4.1) is an inference procedure to relate this class of knowledge with other classes containing more abstract knowledge units. In future versions, this link will be substituted by a call to a rule base or hypothesis engine, which manages relationships between classes as though they were causal hypotheses. In the actual version, the intelligent link contains a search procedure for key words in other classes ('Differentiation', 'Tension', etc.).

Once this procedure finds the successive unit (in the figure 'Social Rivalry' or 'Social Differentiation'),

direct links between the new units are explored. The system may use at this stage the initial global variables to activate new cards (see the connections between 'Iconographic Homogeneity' and 'Privileged Social Roles order', for instance). Some of these cards contain conditional forks that can only be solved using that information. For example, only if the existence of Iconographic Heterogeneity in the data set has been determined will the system activate the card 'Appearance of Social Elites'

The last card is an empty text field, which displays the last state of the working memory. All information is organized now in three new global variables 'Social Conflict', 'Unbalanced Social Structure' and 'Balanced Social Structure'. In other words, at the end of the session and after displaying the final value of global variables, a new working memory is constructed around these new categories.

Reactivation begins when we use the content in the new working memory to invert the reasoning process. For

example, if the Solution Card contains the piece of knowledge 'Balanced Social Structure', the system searches for the theoretical entity that has sent this message. Once localized the content of this previous card is modified. A new text is displayed, explaining the hierarchy chain that was used in the previous inference. The user may now select to explore an alternative hierarchical chain or to browse the next fork in the first chain.

If the user wants to study the reason for a balanced social structure (first hierarchical fork), the system will display a new card with four alternative inferences:

- (i) Variability among other sociotechnical items not represented in stelae.
- (ii) Variability in Settlement Patterns.
- (iii) Social Differentiation on Work processes.
- (iv) Subsistence Economy.

Each possibility is a direct link to a new Data Description Card. First, the system studies a new set of empirical observations (variability among imported ceramics, hierarchical patterns of settlement, quantitative differences among sites or among graves, etc.), and second it tries to reactivate the concept hierarchy. The intelligent link controlling the input to last card reads the new state of working memory and compares it to the actual content of the card. If the knowledge units activated by the new initial state are the same, then the hypothesis will be validated.

#### 4.6 Conclusions

An Intelligent database is a computer system more developed than a stack of data waiting to be consulted. The general architecture herein described will allow us to use computers as 'intelligent' assistants, managing the enormous set of explanatory concepts needed to accomplish scientific tasks.

The specific uses of Intelligent databases are:

- to *use* a theory to explain the meaning of some experimental results. Each experimental result is defined as the initial state of a specific scientific problem (defined by the user, and not by the computer). The computer will search it in its active intelligent database (that is to say, into the representation of the theory) if there is or is not a final state or a sequence of states conducive to this goal. For example, in archaeology we can use intelligent database to discover the social or cultural function of artifacts (middle-range theories).
- to *validate* a hypothesis. There are two possibilities: the system can be used to find out if some goal (a hypothesis) is 'valid' according to a specific empirical data set. In this case, the intelligent database will use the concept hierarchy to compare all initial states possible using the initial state the user proposes. A second related use is to validate connections between classes in the database. For example, if we assume that in a modern cemetery it is possible to discover a

'Stratified Social Order' we will use these data to study the answer produced by the system.

- to study the *coherence* of a theory by introducing new knowledge units and by comparing the results of the expanded intelligent database to its previous implementation.

Investigations into Intelligent databases are relatively recent. Pioneer work was carried out by Nijssen (1984). Current studies are described in Ullman (1988), Parsaye *et al.* (1989), Murdoch and Johnson (1990), Kerry (1990) and Meersman *et al.* (1990). The prototype described here is based on those studies, although there are some important differences:

- the idea of 'spreading activation': a kind of conditional pattern matching throughout the conceptual hierarchy. This principle comes from Neural Network research (Caudill & Butler 1990).
- the dynamic and active character of knowledge units.
- the *analogy* between the internal architecture of the system and scientific reasoning.

The prototype I have described herein has many limitations. It is not an actual intelligent database because it can solve only a limited number of problems. The conceptual hierarchy is defined in such general terms that it is not very useful in archaeological research. The quantity of knowledge units in ESTELAS is also so reduced that there are no memory allocation problems: a useful Intelligent database has to encompass the greatest and most comprehensive knowledge available in a scientific domain (2000 units? 5000 units?).

A future system is underway (the GLADIUS Project) to solve some of these shortcomings.

#### Acknowledgments

I would like to thank all the staff at the C.N.R.S.-U.P.R. 315 (Paris, France) for their kind hospitality during my postdoctoral work in their laboratory. Special thanks are due to H.P.Francfort, J.C.Gardin and M.S.Lagrange for their valuable support. Advice from Gary Lock, Jon Moffett, and an anonymous reviewer have improved the present version of the paper. Neither they nor anyone else except myself is responsible for the inadequacies that remain. This research project has been funded by a Grant from the Spanish Ministry of Science and Education.

#### References

- BALPE, J.P. 1990. *Hyperdocuments, Hypertextes, Hypermedias*, Paris, Eyrolles.
- BARCELÓ, J.A. 1989. *Arqueologia, Lògica y Estadística. Un anàlisi de las Estelas Decoradas de la Edad del Bronce en la Península Ibérica*, Bellaterra, Servei de Publicacions de l'Universitat Autònoma de Barcelona.

- BARCELÓ, J.A. in press "Una Interpretaciòn Socioeconòmica del Bronce Final en el Sudoeste de la Península Ibérica", *Trabajos de Prehistoria*.
- BIELAWSKI, L. & R. LEWAND 1991. *Intelligent Systems design. Integrating Hypermedia & Expert Systems Technology*, New York, John Wiley.
- BROWN, D.C. & B. CHANDRASEKARAN 1989. *Design Problem Solving Knowledge Structures and Control Strategies*, London, Pitman.
- BUNGE, M. 1983 *La Investigaciòn Científica*, 2nd Edition, Barcelona, Editorial Ariel.
- CAUDILL, M. & C. BUTLER 1990. *Naturally Intelligent Systems*, Cambridge (MA), The MIT Press.
- CHURCHLAND, P.M. 1989. *A Neurocomputational Perspective*, Cambridge (MA), The MIT Press.
- CLANCEY, W.J. 1985. "Heuristic Classification", *Artificial Intelligence*, 27(3): 289-350.
- COX, B.J. 1987. *Object-Oriented Programming. An evolutionary approach*, Reading (MA), Addison-Wesley.
- FERBER, J. 1990. *Conception et Programmation par Objets*, Paris, Hermès.
- FOSS, C.L. 1988. "Effective Browsing in Hypertext Systems", *User-Oriented Content-based Text and Image Handling*, RIAO 88 Conference, Cambridge (MA).
- HOOKE, C.A. 1987. *A realistic Theory of Science*, New York, State University of New York Press.
- KERRY, A. 1990. *Integrating Knowledge-based and database Management Systems*. Chichester, Ellis Horwood.
- LANGLEY, P., H.A. SIMON, G.L. BRADSHAW & J.M. ZYTKOW 1987. *Scientific Discovery. Computational explorations on the Creative Processes*, Cambridge (MA), The MIT Press.
- LAURIÈRE, J.L. 1986. *Intelligence Artificielle. Resolution de problèmes par l'Homme et la Machine*, Paris, Eyrolles.
- LAYMON, R. 1984. "The Path from Data to Theory", in J. Leplin (ed.), *Scientific Realism*, Berkeley, University of California Press.
- MCKNIGHT, C., A. DILLON & J. RICHARDSON 1991. *Hypertext in Context*, Cambridge, Cambridge University Press.
- MEERSMAN, R.A., SHI ZHONGZHI, KUNG CHEN-HO (eds.) 1990. *Artificial Intelligence in Database and Information Systems*, Amsterdam, North-Holland.
- MURDOCH, S. & L. JOHNSON 1990. *Intelligent Data Handling*, London, Chapman & Hall.
- NEWELL, A. 1973. "Production systems: Models of control structures", in W.C. Chase (ed.), *Visual Information Processing*, New York, Academic Press.
- NEWELL, A. 1980. "Reasoning, problem solving and decision processes: the problem space as a fundamental category", in R. Nickerson (ed.), *Attention and Performance VIII*, Hillsdale (NJ), Erlbaum.
- NEWELL, A. & H.A. SIMON 1972. *Human Problem Solving*, Englewood Cliffs (NJ), Prentice Hall.
- NEWELL, A. & H.A. SIMON 1976. "Computer Science as empirical inquiry: symbols and search", *Communications to the Association of Computing Machinery* 19(3): 113-126.
- NIELSEN, J. 1990. *HyperText & HyperMedia*, New York, Academic Press.
- NUSSEN, G.M. 1984. "From databases to knowledge bases: a technical comparison", in P.J.H. King (ed.), *Databases: State of the Art Report*, London, Pergamon Press.
- PARSAYE, K., M. CHIGNELL, S. KHOSHAFIAN & H. WONG 1989. *Intelligent Databases*, New York, John Wiley.
- PEARL, J. 1985. *Heuristics. Intelligent Search Strategies for computer problem solving*, Reading (MA), Addison-Wesley.
- PITRAT, J. 1990. *Métaconnaissance*, Paris, Hermès.
- SHRAGER, J. & P. LANGLEY, (eds.) 1990. *Computational Models of Discovery and Theory Formation*, Los Altos (CA), Morgan Kaufman.
- SIMON, H.A. 1973. "The Structure of Ill-structured problems", *Artificial Intelligence* 4: 181-201. (Reprinted in H.A. Simon 1977, *Models of Discovery*, Dordrecht, Reidel)
- SIMON, H.A. 1979. "Information-Processing Theory of Human Problem Solving", in W. Estes (ed.), *Handbook of Learning and Cognitive Process*, 5, Hillsdale (NJ), Lawrence Erlbaum.
- SIMON, H.A. 1983. "Search and reasoning in problem solving", *Artificial Intelligence* 21: 7-29.
- TELLO, E.R. 1989. *Object-Oriented Programming for Artificial Intelligence. A guide to tools and system design*, Reading (MA), Addison-Wesley.
- THAGARD, P.R. 1988. *Computational Philosophy of Science*, Cambridge (MA), The MIT Press.
- ULLMAN, J.D. 1988. *Principles of Database and Knowledgebase Systems*, Rockville (MD), Computer Science Press.
- WAERN, Y. 1989. *Cognitive Aspects of Computer Supported Tasks*, New York, John Wiley.