IKBS : An Iterative Knowledge Base System for improving description, classification and identification of biological objects.

Noël Conruyt, D. Grosser, H. Ralambondrainy

IREMIA
University of La Réunion
15, av. René Cassin - 97715 Saint-Denis
Messag. Cedex 9, France

Abstract

Knowledge bases in natural sciences are made of the description of complex objects, mostly structured and imprecise. In order to get solid results concerning classification and identification, we need appropriate computing methodologies and software in order to describe, classify and identify these symbolic objects.

The IKBS project will produce an integrated multi-platform software workshop available through Internet to build such knowledge bases. It will reinforce the scientific methodology in biology (experimenting and testing) with the appropriate abilities to acquire a descriptive model and collect cases, process those complex descriptions for classification and identification help, and refine all these experimental knowledge.

This methodology has been successfully applied to a subset of the sponge domain (genus *Hyalonema*, 125 cases, 25 objects, 46 attributes). It shows the central role played by the descriptive model, and the necessity for the expert to be able to tune the system for better results.

IKBS will use the Java programming language and will be interfaced with Object Oriented Database System (0_2). Some software components (learning systems, descriptive model editor, questionnaire generator) are already implemented.

Keywords

iterative knowledge base, complex description, classification, identification, natural sciences.

Introduction

In life sciences, computers offer precious help to store, manage and distribute expert knowledge. For example at Reunion island, there exist lots of scientific data in biology that need to be rationally exploited: collections of specimens (herbarium, corals), monographies of species, images, etc.. Knowledge bases are a way to help specialists describe, classify and name taxa (species, genus, etc.). They provide also an identification help for non-specialists who want some information on a specimen whose name is unknown.

Nevertheless, taxa and specimens are often difficult to represent on a computer because of their intrinsic complexity which is hard to describe: polymorphism (e.g. intra-specific variation), dependency relations between attributes, objects and values (related to the organism's structure), and the difficulty to get all the required information when consulting the knowledge base (incompleteness of samples to identify).

These natural characteristics require new tools in order to represent and treat objects[1] that are possibly multiple (several values for sets instead of an average and unique value), inapplicable (this information is different from unknown "values") and fuzzy (thus expressing some uncertainty in user choices).

Questions and illustrations chosen by the expert for observing biological objects cause another kind of difficulty for descriptions. In fact, errors of interpretation often appear when consulting questionnaires. This is partly due to the lack of explanations from the expert (e.g. where to observe ?), to the bad definition of values (e.g. are they mutually exclusive ?), or to the lack of frame of reference (e.g. what is thick and thin for the diameter of spines ?). It is important to record the causes of these "noises" in order to define relevant characters (easy to observe), to offer non ambiguous values (easy to interpret), and so improve descriptions.

At last, the need to represent, manage and distribute expert knowledge in knowledge bases is based on the application of an experimental approach. The latter one is divided into four steps: the observation of facts, the building of hypotheses, testing, and iteration [POP 91].

Taking into account these particularities has led us to conceive a new software workshop: IKBS. It lies on three development viewpoints to represent knowledge:

- conceptual aspects: this step is semantic and deals with the realisation of the predefined objectives with an appropriate methodology,
- formal aspects: at this stage, a language matching the above requirements is defined,
- computerized aspects: the system whose architecture applies the iterative approach, and algorithmes for handling complex description representations and processes.

IKBS conceptual aspects

Methodological issues :

We are currently designing an Iterative Knowledge Base System to build knowledge bases in natural sciences. IKBS reproduces the following knowledge management cycle to extract knowledge (Fig. 1). Two types of knowledge, contextual information and know-how, are distinguished. The former represents encyclopedic and textual data. It provides a lot of information on specimens (ecobiological, geographical, etc.) that can be useful for classification and identification. The latter results from observing specimens *in situ* and *on the table*. The formalisation of such practises gives rise to coded and named data (i.e. descriptions) which form the kernel of the knowledge base.

The term knowledge is controversial in computer sciences [KOD 97]. We give here a practical definition for our purpose in biology: We consider that there are three kinds of knowledge: prior, instanciated and derived. **Prior knowledge** (background or domain knowledge) relates to the definition of what is *observable*, i.e., build a descriptive model which corresponds to the modelisation of data, or metadata [DIE 91]. **Instanciated knowledge** refers to the description

[1] We define objects as structural components of a specimen (e.g. body, spine, etc.).

of *observed* things (data, descriptions, cases, examples). **Derived knowledge** can be compared with produced hypotheses (clusterings, decision trees, rules, identification) discovered from prior and instanciated knowledge. Obviously, knowledge is also grounded in expert's mind and what is formalised ("extracted") is a very small part of his/her experience.

In Knowledge Discovery methodology, knowledge is viewed as a result (output) of a linear process of handling data (input) [FAY 96]. As far as we are concerned, we emphasize a more general interpretation of knowledge which is both input (prior and instanciated) and output (derived). This knowledge will be extracted from a cyclical process, divided into three parts:

Prior and instanciated knowledge acquisition:

- Acquiring a descriptive model by defining all potential or observable objects (composite, rule dependant, specialised), typed attributes and values of the domain,
- Collecting pre-classified and observed (cases) in varied or imprecise states (objects and values), though an automatically built-in questionnaire matching the descriptive model,

Prior and instanciated knowledge processing:

- Processing those cases to learn fuzzy inductive hypothesis (classifications), and testing them with new observations for identification purposes (by determination with rules or by comparison though case-based reasoning),

Knowledge refinement:

- Validating the knowledge derived from descriptions: the expert will be able to update the knowledge base according to the results obtained during classification and/or identification, and thus improve his descriptive model and case base iteratively.

Knowledge acquisition and knowledge processing are two disconnected parts in our method: we consider them to be really independent phases in the process of knowledge management. The aim of knowledge acquisition is to produce descriptions (the extension of classes). The aim of knowledge processing is to deliver useful information to classify or identify these descriptions (e.g. characterisation of classes).

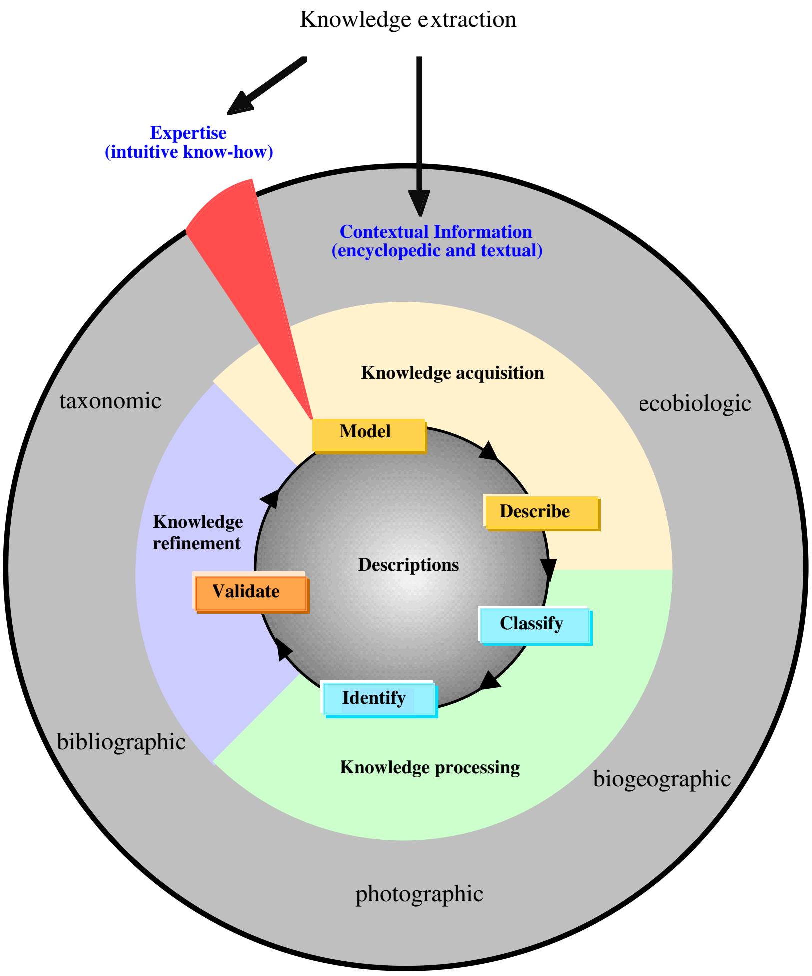

Fig. 1: The knowledge management cycle.

Software issues :

Some software engineering and artificial intelligence programs try to cover this need with a spiral conception method: the validation process takes into account returns from experts in order to improve their qualitative models elaborated in a first step, and to progress in the analysis of problems and definition of methods to solve them.

But the problem we are faced with in biology is more difficult: we have to modify the initial descriptive model, and maintain old data consistent with the new structure. This **iterative** requirement is the **first objective** of our software workshop.

The need for an Iterative Knowledge Base System comes from the difficulty for any specialist of a biological domain to define observable objects, attributes and values that are well suited at once for describing natural individuals. The so called descriptive model is the foundation of the elaboration of a solid knowledge base in natural sciences [CON 97]. But its conception is not a trivial task because it closes the world of possibilities and so must anticipate what will be observed in reality.

For helping the expert in this task, we have set up logical rules for describing specimens in a descriptive model: composition, point of view, specialisation, iteration, contextual conditions, etc. [LE R 96]. The structuration is a means to take into account “good sense” background knowledge that can be useful for description capture, management and processing [ALL 84]. Difficulties of representation come from two aspects:

Definition of the structure of descriptions:

- choice of the status of descriptors (object, attribute, value),
- type definition of attributes (numerical, nominal, taxonomic, ordered or intervals),
- dependency rules between objects (evolution, composition, specialisation, iteration),
- applicability of attributes and values (according to contextual conditions),

Interpretation of descriptors:

- unique object (singular) or multiple objects (set),
- monosemic attributes,
- mutually exclusive values.

For example, if the name of the object covers one representative (e.g. body of the sponge), we can without doubt interpret the answer "cone, trumpet" concerning the values of the attribute "shape" to express uncertainty. But if this name makes reference to a set of representatives (e.g. calices of a coral's specimen), a multiple choice of values should be interpreted as a conjunction of variation (e.g. shape circular+sub-circular) instead of a disjunction of imprecision (e.g. shape circular, sub-circular).The former representation is true information and must not be confused with the latter representation which is a partial decision.

HyperQuest is the module of IKBS that generates complex description trees. Each description is an instance of the descriptive model. By the way, they are comparable with one another (Fig. 2).

The **second objective** is to **handle** such **complex descriptions** in the processing phase. In most of present KDD tools, data are expressed in matrices with cells containing only one value (the average). Moreover, the use of KDD tools presupposes the independance of the descriptors (attributes of characteres). This assumption is not verified in natural domains and we must develop programs able to process such richness in data.

Figure 2 shows the structured description of a specimen of *Cyliconema* (sub-genus of *Hyalonema*). All the information about this specimen was not accessible in reality: for example, the object "peduncle" is dimmed because this part of the specimen was cut when dragging it out from the sea. Some attributes could not also be given an answer (e.g. state unknown for consistency of the body). Apart from taking into account the inapplicability of some attributes (e.g. due to missing central cone), the complexity of handling these descriptions comes from structured attributes (e.g. shape of body), varying ones (e.g. size of amphidiscs), multiple descriptions objects (e.g. there are several kinds of amphidiscs in this description; here, only the first instance is displayed, and other instances can be showed by clicking on the symbol "1>").

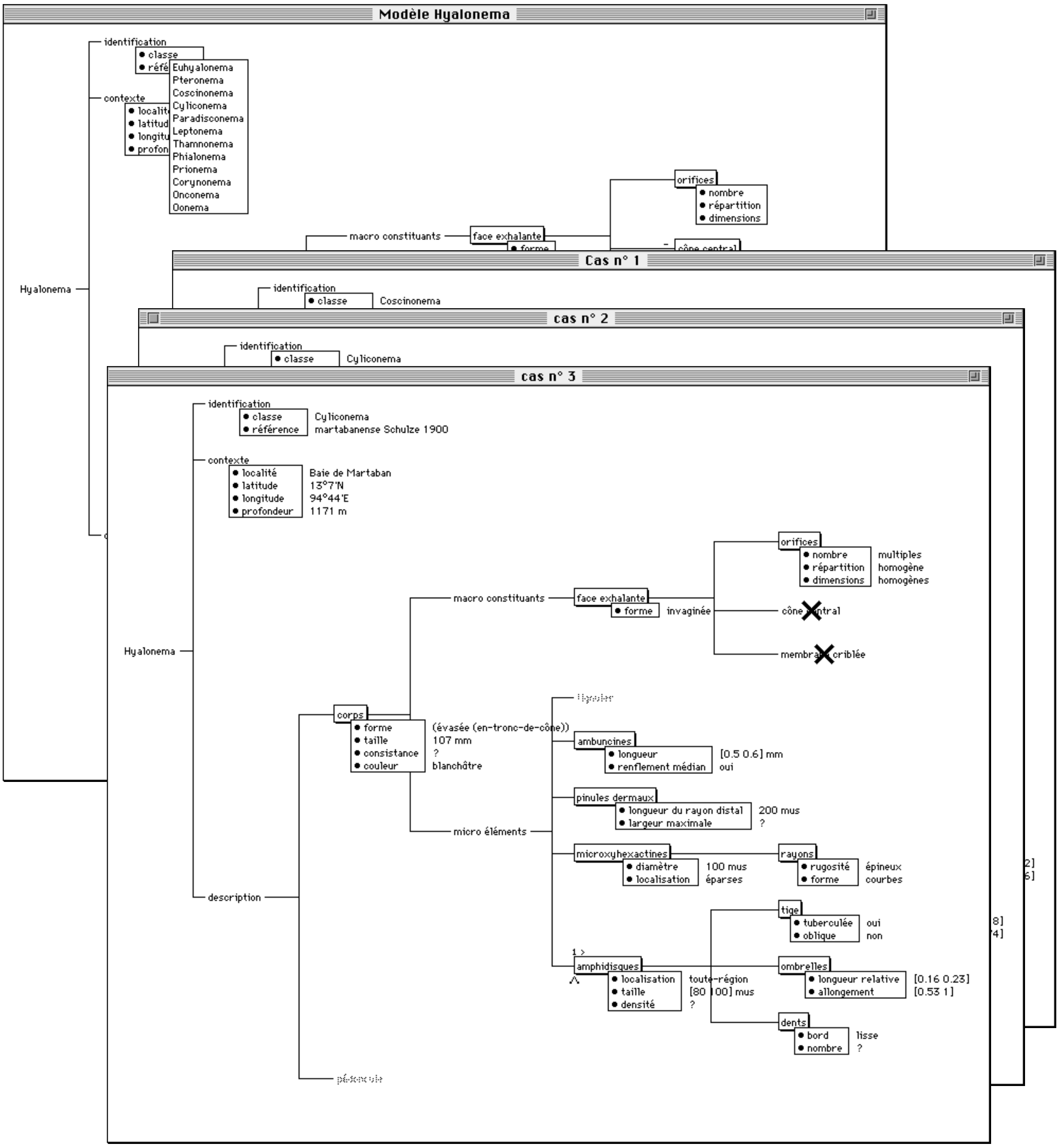

Fig. 2: Description trees of the sponge domain

In HyperQuest we stressed an important point consisting in giving a graphical display of each description. This capability of dealing with **intelligible knowledge** (prior, instanciated and derived) is the **third objective** of IKBS because it facilitates interpretations for every end users of the knowledge base.To put it in a nutshell, the knowledge will be easier to understand if it is well visualised through a proper designed interface. A scripting and object oriented language, HyperTalk (from HyperCard) was chosen to build the interface of IKBS.

IKBS formal aspect

prior knowledge

Traditional hierarchical or relational DBMS handle flat data and assume that information is complete. The IKBS model has been developed to represent structured data that may includes irrelevant, unknown and imprecise information. An observation from a given set of data is considered as a structured object, i.e., it is made up of components and a each of them may itself be composed (see Fig. 2). Attributes are used to describe objects or components. Components or values may be "unknown", "irrelevant" or "absent", fuzzy values are qualified using linguistics terms such as "more or less", "few", etc..

More precisely, *IKBS formal language* is built in the same way that strong typed languages and object-oriented data model [DEL 91]. We assume that it is given a finite set $\{(A, D)\}$ where A is an attribute identifier and D its domain definition. According to the domain structure, we define a set of basic types: numeric, intervalNumeric, symbolic, fuzzySymbolic, taxonomic, etc..

Then new types can be defined using type constructors such as tuple and set. If $\tau_1, \ldots, \tau_n$ are types then $\tau{:}\langle \tau_1, \ldots, \tau_n \rangle$ is a type called a *tuple type*, and if τ' is a type then $\tau{:}\{\tau'\}$ is a type called a *set type*.

When a tuple type attribute A is constructed using the attributes $A_1, \ldots, A_n$ having the types: $\tau_1, \ldots, \tau_n$, it is denoted: $A{:}\langle A_1{:}\ \tau_1, \ldots, A_n{:}\ \tau_n \rangle$. The attribute A defined as a set of attribute B is denoted $A{:}\{B{:}\ \tau_B\}$. These constructed types are useful to describe structured entities and their components.

For example, the “exhaling face” attribute for the descriptive model of *Hyalomena* is represented by the tuple (see fig. 2):

ExhalingFace:$\langle$

shape:symbolic,

$\langle$RiddledMembrane: $\tau_{riddledMembrane}$,CentralCone: $\tau_{centralCone}$,Orifices:$\tau_{orifices}\rangle\rangle$

One micro element of a *Hyalomena* sponge is called “amphidiscs” and is described as follows:

Amphidiscs:$\langle$

localisation:symbolic, size:intervalNumeric, density:symbolic,

$\langle$Teeth: τ_{teeth},Ombrella: $\tau_{ombrella}$,Stem: $\tau_{stem}\rangle\rangle$

but the “micro element” attribute which is composed of a set of amphidiscs is represented by the set:

Micro-elements: $\langle$

$\langle$Tignules: $\tau_{tignules}$,

Ambuncines: $\tau_{ambuncines}$,

DermalSpines: $\tau_{dermalSpines}$,

Microxyhexactins: $\tau_{microxyhexactins}\rangle$,

$\{$Amphidiscs: $\tau_{amphidiscs}\}\rangle$

This representation allows to deal with multiple instances of the same object.

At last, to take into account fuzziness of values, we fix a valuation set {V} where V is a set of fuzzy qualifiers. In our approach, fuzziness is not expressed as usual by using the interval [0,1], but with chain or lattice structures (see fig. 3).

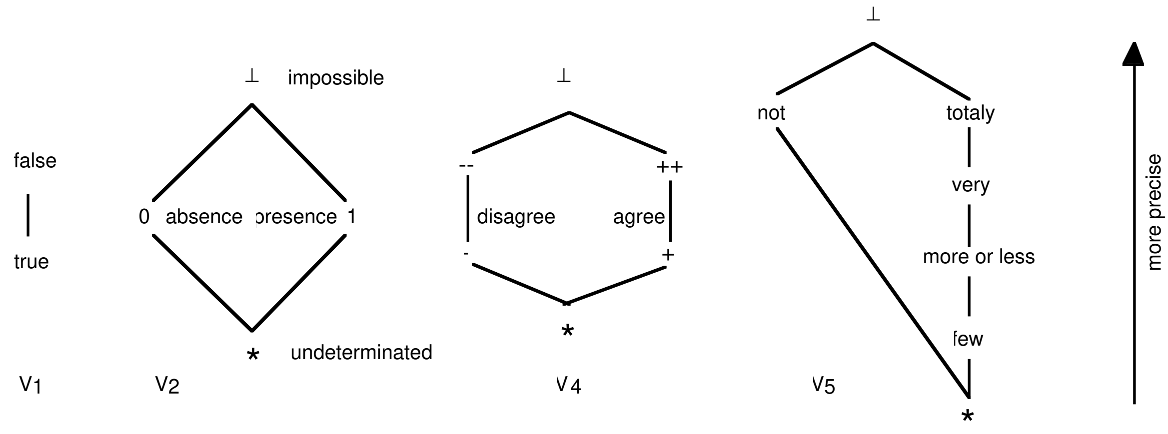

Fig. 3: Some fuzzy qualifier lattices

A value may be more general or more precise than another one. For learning purposes, lattice-fuzzy sets [GIR 97], [LOU 97] give the possibility to qualify imprecise values with different semantics, e.g. V_1 boolean chain, V_2 belief bilattice [GIN 88], V_4 agreement lattice, V_5 intensity lattice, etc.. A qualified value for an attribute A is a function $\mu : D \rightarrow V$ that associates a qualifier in V with each modality d_i in D.

Before using the IKBS system to process data, the user must choose a set of typed attributes $\{A_j{:}\tau_j\}$. Then he must use them to describe the structure of its data. The descriptive model is represented as a tuple attribute:

DescriptiveModel:$\langle A_1{:}\tau_1, \ldots, A_r{:}\tau_r \rangle$

For example, the descriptive model of the *Hyalomena* sponge is the following attribute tuple:

HyalomenaModel:<

Identification: $\tau_{identification}$,Context: $\tau_{context}$,Description: $\tau_{description}$>>

instanciated knowledge

For a precise attribute (A, D), the set of instances or values denoted [A] is the domain D.

If A:$\langle A_1{:}\tau_1, \ldots, A_n{:}\tau_n \rangle$ is a tuple attribute, we have [A] = {?, NA , no} $\cup$ $[A_1]$ x...x $[A_n]$ and for a set type attribute A:{B: τ_B}, we have [A] = {?, NA , no} $\cup$ {[B]}.

For example, an instance of the “body” attribute for case n° 3 of fig. 2 is:

<Body: <shape: (wide-mouthed (truncated-cone))>,

<size: 107>,

<consistency: ?>,
<color: whitish>
<MacroConstituents:
<ExhalingFace: <shape: invaginated>
<RiddledMembrane:no: <distinct: NA>>,
<CentralCone:no: <position: NA>>,
<Orifices: <number: multiple>,
<repartition: homogeneous>,
<dimensions: homogeneous>>>
<MicroElements:
<Tignules: ?: <lenght: ? >>,
<…>
>

For a fuzzy attribute (A,D,V), [A] is V^D.

If $A:\langle A_1:\tau_1, ..., A_n:\tau_n\rangle$ is a tuple attribute, we have $[A] = [A_1]$ x...x $[A_n]$ and for a set type attribute $A:\{B: \tau_B\}$, we have $[A] = \cup \{[B]\}$.

For example, with V_2 belief lattice for all the attributes except for attribute "repartition of orifices" which is V_5, an instance of the "body" attribute for case n° 3 of fig. 2 is:

<Body:1: <shape: ((sub-cylindrical 0) ((bottle 0) (cylindrical 0))
((swollen 0) ((flute 0) (corolla 0) (pyriform 0) (ovoid 0) (bulbous 0))
((wide-mouthed 1) ((truncated-cone 1) (cone 0) (trumpet 0) (crater 0) (cornet 0) (bell 0))
((flattened 0) ((flat 0) (folded lamina 0))>,
<size: 107>,
<consistency: (rigid ∗) (soft ∗)>,
<color: (whitish 1) (greyish 0) (other 0)>
<MacroConstituents:1:
<ExhalingFace:1: <shape: (invaginated 1) (plane 0) (flattened 0)>,
<RiddledMembrane:0: <distinct: (yes ⊥) (no ⊥)>>,
<CentralCone:0: <position: (prominent ⊥) (non-prominent ⊥)>>,
<Orifices:1: <number: (unique 0) (four 0) (multiple 1)>,
<repartition: (homogeneous few) (heterogeneous very)>,
<dimensions: (homogeneous 1) (heterogeneous 0)>>>>
<MicroElements:1:
<Tignules: ∗: <lenght: ∗>>,
<…>

>

derived knowledge

Depending on the goal to be achieved, two different types of method are used in order to process the case base: induction for classification, case-based reasoning (CBR) for identification.

For classification purpose, a decision tree is built. Starting from descriptions (representation in extension) of classes to learn, an inductive method based on entropy and information gain measure [SHA 1949], [QUI 86] gives a characterisation of these classes (representation in intension) with a set of rules. Each path from the root to a leave of the decision tree is a classification rule.

For identification purpose, a CBR strategy is used [BAR 89]. Given a set of examples, it dynamically extracts the most efficient criteria from the ordered list of tests after each answer of the user. The cases are selected from this reply. If the answer is unknown, the second most discriminant test is proposed to the user, and so on.

Even if this method of identification is more resilient to this noise (unknown responses), it is not as good when facing errors of description. This is due to the monothetic approach of this strategy [PAN 91]: it is based on one and only one criterion at a given moment.

Other methods of CBR are polythetic (i.e. rely on a combination of criteria) and are more robust to errors: the most similar cases method is derived from the k nearest neighbours one in data analysis: the examples are retrieved in calculating a similarity measure between descriptions. This is a comparison process which implies the whole set of characters (or attributes). A score between 0 and 1 gives the percentage of resemblance between two cases.

For the consultation, there is an interest to combine these methods (induction and CBR) at different levels of integration in order to get more solid results [AUR 94].

IKBS computerized aspects

Knowledge base development environment

In order to handle previously defined software issues (iterative approach, complexity and intelligibility of knowledge, overture), we need an integrated platform based on the “all object” paradigm. IKBS is a modular system composed of specialized tools for acquiring, processing and validating knowledge items (prior, instanciated, derived). The object-oriented design method is used to conceive IKBS: the software development life cycle using object-oriented design emphasizes the incremental, iterative development of a system, through the refinement of different yet consistent logical and physical views of systems as a whole [BOO 90].

In fig. 4, we illustrate the three above objectives of our platform:

- Complex knowledge representation: our applications have to deal with structured and imprecise data. For example, knowledge discovery tools must take into account dependencies between objects for the learning processes (discrimination, clustering, etc.). Similarly, uncertainty in descriptions should prevent us from taking too much risks in the monothetic approach of CBR for identifying a new observation.
- Intelligibility and overture of knowledge: we want end-users (principally experts and other biologists) to appropriate the different tools, because they are not accustomed to computers. This is why we take care to bring them high quality graphical representations of the knowledge using a multimedia environment. We aim also to give a wide access to knowledge bases: this will be done with web interactive tools. At last, we intend to produce

some inter-operability of IKBS with object oriented databases (O_2), and to translate the knowledge in other representation formats such as DELTA [DAL 93].

- Iterative method: there are evolving versions of the same knowledge base, according to bad or good results of classification and identification. Derived knowledge is submited to the knowledge validation cycle. Intra-validation is the process of evaluating the consistency of the produced knowledge in comparison with instanciated and prior ones in a same version of the knowledge base. Inter-validation is the process of controling objects relations with ancient versions of themselves [SEL 97].

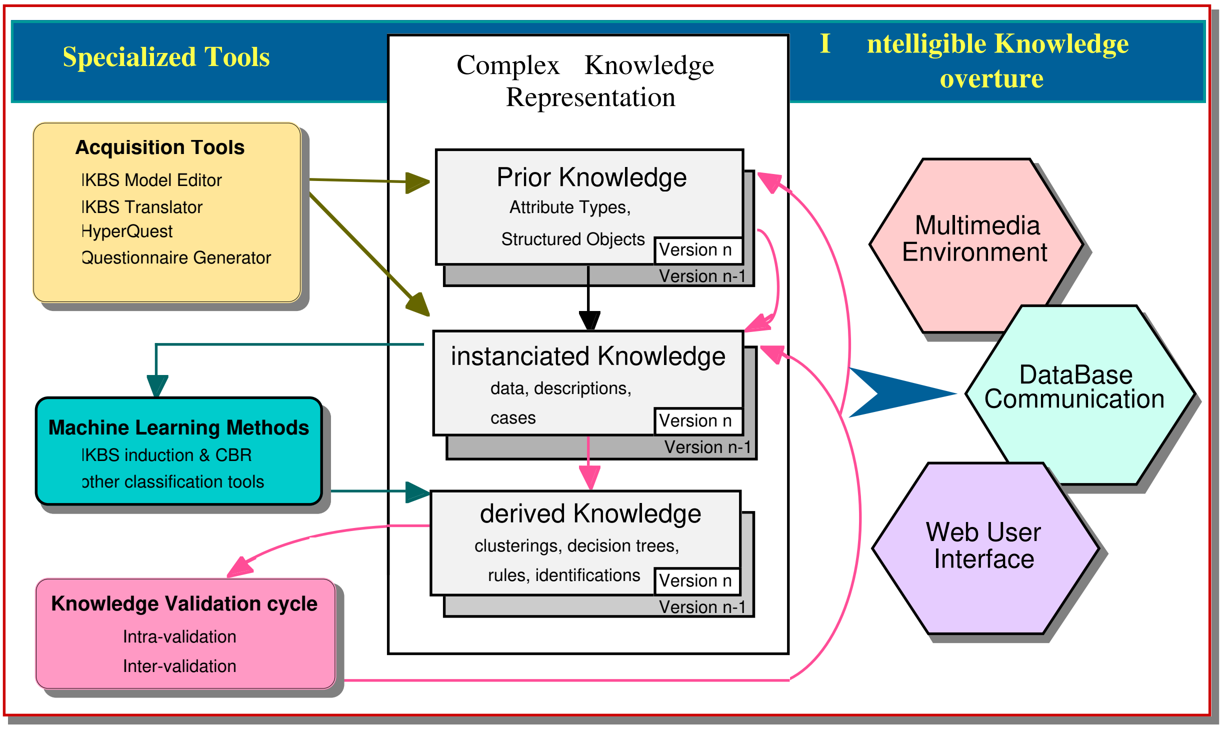

Fig. 4: The knowledge base development environment

The validation process is what justifies the construction of our knowledge base development environment. Until now, most of our work was concerned with the acquisition and processing phases. Today, we concentrate on a crossed validation methodology of observable (prior) and observed (instanciated) knowledge. This phase is currently functional but relies on empirical bases. For example, an experimentation is made which consists of testing case base reasoning methods for identification with the expert and different users. Each one describes the same specimen with the questionnaire. A crossed validation is made with descriptions from one person (test cases) against the remaining ones (case base). The evaluation of results reveals "noisy" attributes, i.e. those which lead to misclassifications, because of different interpretations of what should be observed (from the expert viewpoint).

Architecture

One of the characteristic features of the IKBS system is that it integrates high-level tools that implement the previous knowledge management cycle (see fig. 1). At present, IKBS is developped with three different programming environments : HyperCard, Java and the Object Oriented DataBase O_2 (see fig. 5).

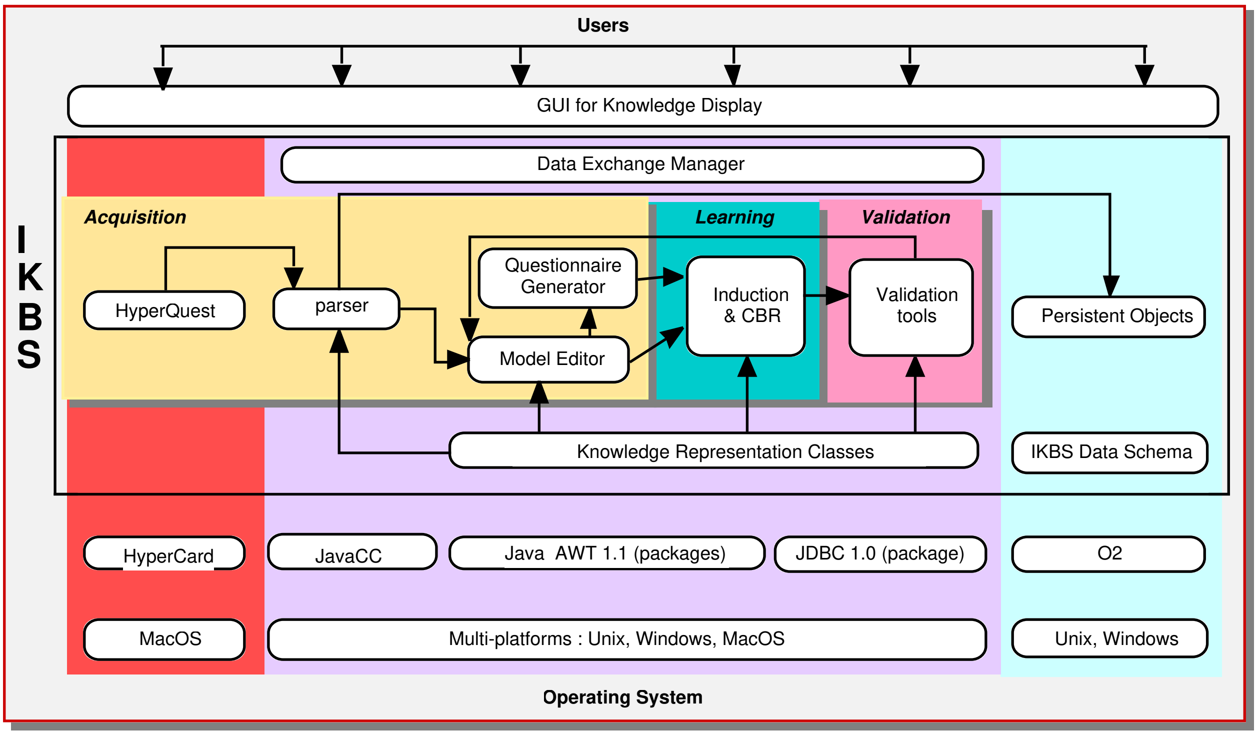

Fig. 5: IKBS architecture

HyperCard environment:

Acquisition tools are built with HyperTalk which is the scripting language of HyperCard.

HyperQuest (a set of HyperCard stacks) is a knowledge acquisition tool with a high level hypertextual and graphical user interface for descriptive knowledge representation. It allows the user to create structured descriptive models (prior knowledge), generates questionnaires that can be used to lead to cases' acquisition (instanciated knowledge) or identify new observations [CON 94]. Its multimedia facilities are useful for illustrating the attributes and make prototypes of knowledge bases.

Java environment:

For the moment, HyperQuest is the front-end knowledge acquisition module of IKBS: it is research oriented and allow to prototype easily human interfaces. In order to be multi-platform and accessible through Internet, a model editor and a questionnaire generator will be written in Java. Thus, the acquisition process can be made by HyperQuest for local use or by the Java Model Editor and Questionnaire Generator for network use. The communication with HyperQuest is realized by a parser written in JavaCC, a Java parser Generator (like Lex and Yacc on Unix System), which translates HyperQuest data into an internal Knowledge Representation Language (KRL). Java classes implement the formal IKBS language defined in the previous section.

As shown in fig. 5, each IKBS tool includes the knowledge representation classes as a java package to communicate one another. The integration of tools is realized by a Data Directed Model implemented with a Data Exchange Manager layer. A Data Exchange Manager (DEM) is a framework to integrate isolated tools by coordinating and controling data exchanges between them [BEN 96].

O_2 environment:

When working with a lot of cases, we would like to manage the persistence of objects in an object oriented langage. We defined a Data Schema on the Object Oriented Database O_2 which is formally equivalent to the KRL defined previously. The main task of the DEM consists in managing data exchange between the KRL and the Data Schema through the JDBC (Java Data Base Connectivity) interface. Java Database Connectivity is a standard SQL/OQL database access interface, providing uniform access to a wide range of relational and object oriented databases. In order to take into account the user needs (to view and to manipulate different versions of the same object), we consider to use a multiversion database formal model where both objects and database schema may be multiversioned. A database version comprises one version of each multiversion object and class that are bound together to represent consecutive states and state variants of the entire real world modeled that may vary by their values, structure and behaviour [JOM 91].

Conclusion

In this paper, we argued on some essential methodological aspects to extract expert knowledge in natural sciences. To catch the complexity and the constant evolution of this knowledge, we proposed a Knowledge Management Cycle as a solution for dealing with three knowledge levels: prior, instanciated and derived. Then, we defined a formal language based on strong typed languages and object-oriented data model to represent them. At last, we illustrated the main characteristic features of the IKBS system: iterative refinement, representation of complex descriptions, intelligibility and overture. This is realised with a mixed architecture developped from different programming environments : HyperCard, Java and O_2.

Experiencing this methodology on the sponge domain showed clearly that this approach is very attractive to express descriptive knowledge and to perpetuate it in an evolving knowledge base. IKBS is currently tested on corals (family *Pocilloporidae*, about 100 cases, 37 objects, 87 attributes). After evaluation of identification results with a first descriptive model and case base, we are refining them through the iterative step. For the moment, this process is essentially manual (empirical). Nevertheless, we intend to find amendment rules to update automatically this old version of descriptive model and associated case base.

In order to gain more robustness in the use of knowledge bases, the definition of a formal model of these rules is the next unavoidable step to clear.

References

[ALL 84] Allkin R., *Handling taxonomic descriptions by computer*, Allkin R and Bisby FA (Eds.), Databases in systematics. Systematics Association London, Academic Press, (26): 263-278, 1984.

[AUR 94] Auriol E., Manago M., Althoff K.D., Wess S., Dittrich S., *Integrating induction and case-based reasoning: methodological approach and first evaluations* EWCBR-94 - Second European workshop on case-based reasoning. M Keane, JP Haton & M Manago (Eds.), AcknoSoft Press, pp 145-155, 1994.

[BAR 89] Bareiss R., *Exemplar-based knowledge acquisition: a unified approach to concept representation, classification and learning*, London, Academic Press inc, 1989.

[BEN 96] Benadjaoud G.N., DEE : un Environnement d'Echange de Données pour l'intégration des applications, Thèse de doctorat en informatique, Ecole Centrale de Lyon, 1996.

[BOO 90], Booch G., *Object Oriented Design with Applications*, Benjamin/Cummings Publishing Company, Inc, 1990.

[CON 94] CONRUYT N., *Amélioration de la robustesse des systèmes d'aide à la description, à la classification et à la détermination des objets biologiques*. Thèse de doctorat en informatique, Univ. Paris-IX Dauphine, pp 1-281, 1994.

[CON 97] CONRUYT N., GROSSER D., FAURE G., *Ingénierie des connaissances en Sciences de la vie: application à la systématique des coraux des Mascareignes*, Journées Ingénierie des Connaissances et Apprentissage Automatique (JICAA'97), Roscoff, pp.513-525, 1997.

[DAL 93] DALLWITZ M.J., PAINE T.A., ZURCHER E.J., *User's Guide to the DELTA System: a General System for Processing Taxonomic Descriptions*, 4th edition. (CSIRO Division of Entomology), Canberra, 1993.

[DEL 91] DELOBEL C., LÉCLUSE C., RICHARD P., *Bases de Données : des systèmes relationnels aux systèmes à objets*, InterEditions, Paris, 1991.

[DIE 91] DIEDERICH J.R., MILTON J., *Creating domain specific metadata for scientific data and knowledge bases*, IEEE Trans., Knowledge Data Engineering 3(4): 421-434, 1991.

[FAY 96] FAYYAD U., PIATETSKY-SHAPIRO G., PADHRAIC S., *From Data Mining to Knowledge Discovery in Databases*, AI magazine, 17(3): 37-54, Fall 1996.

[GIN 88] GINSBERG M.L., *Multivalued logics: A uniform approach to inference in artificial intelligence*. Computational Intelligence, 4:265-316, 1988.

[JOM 91] JOMIER G., WOJCIECH C*., Formal Model of an Object-Oriented Database with Versioned Objects and Schema* , Université Paris-Dauphine, 1991.

[GIR 97] GIRARD R., RALAMBONDRAINY H., *The CID system for Concept Acquisition*, Second International ICSC Symposium on Fuzzy Logic and applications, ISFL'97, ICSC Academic Press, pp.320-326, 1997.

[LE R 96] LE RENARD J., LEVI C., CONRUYT N., MANAGO M., *Sur la représentation et le traitement des connaissances descriptives : une application au domaine des éponges du genre Hyalonema*, vol. 66 suppl., Biologie, Recent advances in sponge biodiversity and documentation, P. Willenz (Ed), Bulletin de l'Institut Royal des Sciences Naturelles de Belgique, pp. 37-48, 1996.

[LOU 97] LOUTCHMIA D., RALAMBONDRAINY H, *Inductive Learning using similarity Measures on L-fuzzy set*, 6th IEEE Int. Conference on Fuzzy Systems, Barcelona, 1997. (to be published).

[PAN 91] PANKHURST R.J., *Practical taxonomic computing*. Cambridge University Press, Cambridge, pp 1-202, 1991.

[POP 91] POPPER K.R., *La logique de la découverte scientifique*. Payot (Eds.) Press, Paris, 1973.

[QUI 86] QUINLAN J.R., *Induction of decision trees*, Machine Learning 1 : 81-106, 1986.

[SEL 97] S ELLINI F., YVARS P.A., *Modèles de connaissances dans les SBC : aller vers une méthode générique de validation,* Journées Ingénierie des Connaissances et Apprentissage Automatique (JICAA'97), Roscoff, pp.448-451, 1997.

[SHA 49] SHANNON C.E., *The mathematical theory of communication*, University of Illinois Press, Urbana, 1949.