Techniques of Engineering Applied to a Non-structured Data Model

Cristóbal J. Carmona[1], María J. del Jesus[1],
Pablo Guerrero[2], Reyes Peña-Santiago[2], and Víctor M. Rivas[1]

[1] Department of Computer Science and [2] Department of Animal Biology, Plant Biology and Ecology, University of Jaen, Campus Las Lagunillas s/n, 23071, Jaen
{ccarmona,mjjesus,pguerre,rpena,vrivas}@ujaen.es

Summary. The information developed pertaining to biodiversity studies tends to be scattered around many bibliographic references. A review of the Nordiidae family is being carried out, but the very data to be collected does not allow systematic access. The Nordiidae family, belonging to the animal taxon Nematodes, shows high diversity and an extraordinary ubiquity. Engineering techniques have been applied to turning this textual information into structured data, so that new knowledge can be discovered and data can be accessed through the net.

Keywords: Data Engineering, Automated Processing, Hypermedia System, Regular Expressions Extraction.

1 Introduction

One of the most important limitations in biodiversity studies is that the available information is dispersed throughout many bibliographic references, and frequently structured under different criteria. A revision of the family Nordiidae is currently been carried out by the Andalusian Group of Nematology, compiling the available information about the most representative genus and species. This updated information has been stored in a large number of unstructured documents. This work shows a software engineering procedure developed to translate this series of text documents into structured data, in order to improve their diffusion within the scientific community.

Nematodes (phylum Nematoda or Nemata) are an animal taxon showing high diversity [7] and extraordinary ubiquity, despite their simple morphological body plan [4]. The study of soil nematodes has received much attention during the last decades as they cause severe diseases in cultivated plants, meanwhile many free-living species are good (bio)indicators of soil quality (health). A set of text documents compiling available information of *Enchodelus* species has been prepared to describe and clarify their taxonomy. The information of each species consists of the following items:

1. **Nomenclature:** scientific name, and its binomen: authorship and date.
2. **Synonymy** (if it exists)**:** other synonymous scientific names and their corresponding bibliographic references.

J.M. Corchado et al. (Eds.): DCAI 2008, ASC 50, pp. 410–414, 2009.
springerlink.com

3. **Description:** morphological and morphometric data for both female and male, when available.
4. **Diagnosis:** a brief report of useful characteristics for species identification.
5. **Relationships:** a comparison of each species with its nearest relatives.
6. **Distribution:** data referring to the geographical distribution of the species.
7. **Type material:** number of type specimens and collections and/or places where they are deposited.
8. **Remarks:** additional comments on the species in question.

The complete procedure for turning the unstructured data into structured is detailed in the following steps: Section 2 describes the structure of the database storing the information. In Section 3 the different procedures used to generate structured data are listed. The final section shows conclusions and future prospects for this project.

2 Analysis of the Classification Structure

The revision of the family Nordiidae generated a series of documents to which database techniques have been applied. Thus, all the information is stored in a structured and easily accessible way.

Figure 2.a shows an example of a text document compiled from the original references. It includes taxonomic information (genus, sub-genus and species), authority, synonyms, and a list of bibliographic references. These references are grouped according to the name used by each author. After the nomenclature has

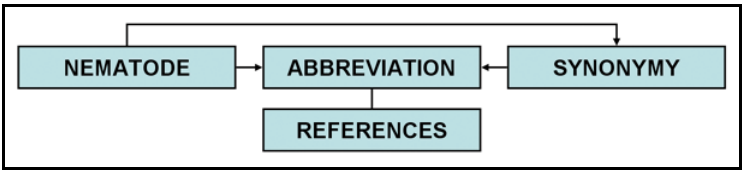

Fig. 1. Relational model for the system

Table 1. Tables generated from the revision with primary key indicated

NEMATODE(genus, species, subgenus, author_gen, year_gen, author_sp, year_sp, female, male, diagnosis, relationships, distribution, etymology, type_material, remarks, cod_icon)
SYNONYMY(gen_syn, sp_syn, gen_nem, sp_nem, subgenus, author_gen_syn, year_gen_syn, author_sp_syn, year_sp_syn)
ABBREVIATION(author, year, gen_nem, sp_nem, gen_syn, sp_syn, subgen_nem, subgen_syn,author_gen_syn, cite)
REFERENCES(id, author, year, nick, cite)

been established, the morphological features seen in section 1 are described (for instance, *Female*, *Male*, of *Diagnosis*)

The entity-relation diagram (shown in Figure 1) is used to obtain the normalized tables that will store all this information, in order to make possible its manipulation and visualization in different media. Normalized tables are shown in Table 1. Thus, NEMATODE is the main table and stores the valid name of the species together with all its features. The SYNONYMY table includes all synonymous names the nematode has received. The ABBREVIATION table stores all abbreviations for the references used in the original document. Finally, the REFERENCE table contains a full description of the references.

3 Data Processing Method

Introducing information into the database is a time-consuming task prone to mistake, due to the characteristics of the original text documents. Thus, data cannot be extracted in a straightforward way since fields are described by means of differences in style labels, and these labels are expressed and saved in an specific and proprietary wordprocessor format, making their processing quite difficult.

The automatic data extraction process is described in the following two sections: Firstly (see section 3.1) the justification for using a semi-structured document and the processing carried out in order to obtain this kind of document from the original. Secondly (see section 3.2) the procedure which obtains the items of the database from the semi-structured document.

3.1 Original Document Processing

A Relational Database can be considered as a type of semi-structured[1] document [8] considering the following reasons:

- It has a series of requirements very similar to those of a relational document.
- Values can be organized in the same way as in a database.
- Just as in a Relational Data Base, the attribute of a register may be another register in semi-structured documents.

Following the philosophy developed by Chris Bizer in *D2R-Map* [2] and *D2R-Server* [3] and Pérez de Laborda in *Relational.OWL* [5], where they transform a database into a semantic document (RDF), the data processing begins with the conversion of the original document into a semi-structured document. By means of word-processing tools, the un-structured document is transformed into an XHTML document, which is still complex to process since the branches generated rarely contain the same number of nodes.

[1] Semi-structured Data - http://ict.udlap.mx/people/carlos/is346/admon07.html

3.2 Semi-structured Document Processing

Since our original documents distinguished fields according to style labels, XHTML seemed to be the best solution for temporarily storing the information. In effect, XHTML allows the inclusion of style labels in the processed document. In this way, each different style identifies certain features, such as genus, species or author, which can be associated to the database fields. This association allows the identification of fields by means of detecting style patterns. In figure 2.b for instance, the *T1* style identifies the genus and species of the nematode, the *T2* style identifies the author of the species and so on. As values for fields are being obtained, the SQL insertion commands for a MySQL database are simultaneously generated.

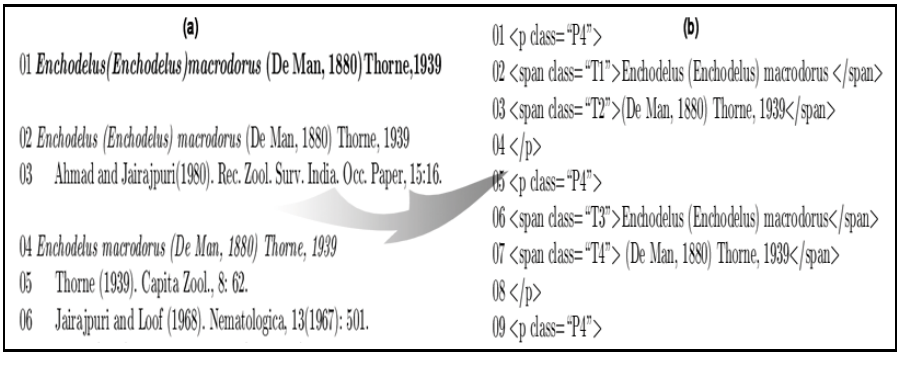

Fig. 2. Original(a) and Semi-structured(b) document

The implementation is carried out in Java, since many free packages and components can be used in order to perform a pattern search. It is thus possible to complete the conversion from semi-structured document to items-insertion document in a short time. The algorithm developed for this project includes pattern recognition systems for finding links to other species or authors present in each field of the database. Due to implemented the highly variable bibliographic patterns of the original document, a complex process to extract regular expressions has been implemented. For instance: from "*author, year1, year2*", the process obtains two distinct links "*author year1*" and "*author year2*". Furthermore, the algorithm is designed to maintain the initial format when presented to the user. This processing is carried out using the following Java regular expression:

$$(\backslash\backslash([A-Z][a-z]+(-|\backslash\backslash s|,|[A-Z][a-z]+|and|\&)+\backslash\backslash d\{4\}(,\backslash\backslash s\backslash\backslash d\{4\})*\backslash\backslash)$$

4 Results and Conclusions

At the moment, access to the data pertaining to the Nordiidae family is achieved by means of a web page[2] that includes a sample revision of genus *Enchodelus*. The

[2] http://www.ujaen.es/investiga/nmundii

hypermedia system generated using engineering techniques allows us to access other species when referred to in the text by their valid names or synonyms. Cross references make it possible to navigate through the information of every species, visualizing the existing relationships among them.

Current work includes the development of a powerful system to perform complex searches. This will lead to an intelligent search system based also on frequent search patterns.

Accessibility to data will be also improved by means of *Ajax* [1][6], and also web services. The latter will help to develop useful tools able to join information from different sources, such as genetic and scientific publication databases.

Acknowledgments

This work has been supported by the project RNM-475, of the Consejería de Innovación, Ciencia y Empresa, Junta de Andalucía, Spain.

References

1. Babin, L.: Beginning Ajax with PHP. Apress (2007)
2. Bizer, C.: D2R Map - A Database to RDF Mapping Language (2003)
3. Bizer, C., Cyganiak, R.: D2R Server - Publishing Relational Databases on the Semantic Web (2004)
4. Brusca, R.C., Brusca, G.J.: Invertebrados, p. 1005. McGraw-Hill Interamericana, Madrid (2005)
5. de Laborda, C.P., Conrad, S.: Relational OWL. In: Hartmann, E. S., Stumptner, M. (eds.) Second Asia-Pacific Conference on Conceptual Modelling (APCCM 2005), vol. 43 (2005)
6. Hadlock, K.: Ajax for Web Application Developers. Kindle Books (2006)
7. Hawksworth, D.L., Kalin-Arroyo, M.T.: Magnitude and Distribution of Biodiversity, p. 1140. Global Biodiversity Assessment: 107-191 (1995)
8. Maier, D.: Theory of Relational Databases. Computer Science Pr. (1983)