

Developing Intelligent Systems for Library Works and Services using Declarative Language: a Case Study of Library Classification

Dr Pijushkanti Panigrahi, Dept of Library and Information Science, University of Calcutta, 87/1 College Street, Kolkata, PIN – 700 073, West Bengal, India (e-mail : panigrahipk@yahoo.com)

Abstract

This research article attempts to build an intelligent system for library activities specially library classification in the automated environment. Basically this approach is highly interdisciplinary one encompassing Natural Language Processing (NLP) and Expert Systems (ES), the branches of Artificial Intelligence (AI). An overview on computer programming languages mentioning the advantages of declarative language over the procedural language in respect of AI based programming is given. The usage of logic based representation of knowledge in the field of Library and Information Science with a special reference to library classification is mentioned. Describes the procedure of logic programming mentioning the reason of choosing PROLOG (Programming in Logic), a declarative language, for developing knowledge base and rules for expert systems. Mentions also the reason of using first order predicate logic (FOPL) for this purpose. A new line of thinking is introduced for betterment of the library works and services. Discusses the importance of declarative programming language for developing AI based system. Shows the method of declaring facts in the form of frame and developing rules for constructing classification number using to Prolog code. A small knowledge base and a few rules as sample for developing an automatic library classification system and its functioning techniques are demonstrated as case study of analytico synthetic classification.

Keywords : *Analytico Synthetic Classification, Automatic Classification, Declarative Language, Logic Programming, Prolog, NLP, Knowledge Based Computer System.*

1 Introduction

Any system may be believed to be an intelligent system if it can prove its capability of replicating the human intelligence that minimizes the distinction between man and machine as far as possible. A Knowledge Based Computer System (KBCS) otherwise known as Expert System (ES) in the area of Artificial Intelligence (AI) behaves as an intelligent system. Such systems can be developed through computer programming, which shows its intelligent behavior like human being to some extent. This AI based programming goes beyond the conventional computer systems and can demonstrate a high level of intelligence. Such AI programming is done using computer programming languages. Besides the heard high level programming languages, there are hundreds of languages for use of both in general and specific purposes. Also many new languages are coming up, although many of them may not get popularity in future. However, the recent literatures on programming have introduced many new concepts and classification, often with blurred distinctions and overlapping, e.g. conventional – non-conventional, deterministic – non-deterministic, procedural – declarative, imperative - declarative, a few to mention here [1]. Of all these groupings, special characteristics of declarative language lead to its use for developing of AI based intelligent system for library activities.

1.1 Declarative Language: advantages

Procedural languages emphasize on ‘How’ a particular operation is performed i.e. the program states clearly the procedure to achieve a goal. This is the reason why it is also named as “procedure oriented languages”. On the other hand, in a declarative language, the emphasis would be on ‘What’ is to be computed irrespective of particular method adopted for computation [2]. Declarative languages are based on declaration of facts and rules are written on the basis of ‘what’ is to be computed or inferred. It is needless to state that declarative language is used for developing AI based intelligent system following the logic programming techniques. Because, a set of facts are to be declared logically; and rules are to be developed in the ways of human thinking procedure to some extent. Hence, properties of declarative language are much useful for the present case study. Logic programming is done more comfortably by declarative language, as opposed to more traditional imperative languages such as C, Pascal etc [3].

2 Faceted Classification

Of all activities, Classification has an important role in library works and services. There exists a powerful CoSSCo (Coordinate, Superordinate, Subordinate and Collateral) relationship among all members of the universe of subjects. This relationship is also called as hierarchical relationship. This proves that the nature of the universe of subject is non-linear or multi-dimensional. But arranging documents in the shelves must be linear in nature, as users feel convenient to move in a line while browsing. The library classification forces to change the non-linear nature to linear nature when documents are displayed on the shelves. In other words, the library classification brings related subject groups together and unrelated ones be separated from others. Thus, library classification is meant for arrangement of documents in order to bring like documents together, following the APUPA pattern (Alien, Penumbra, Umbral, Penumbra and Alien). The purpose of arranging documents in this pattern is to facilitate effective browsing to users, who may be interested, not only the documents of umbral region but also of penumbra region.

Library classification has come a long way from early days of enumerative systems. It is Ranganathan, who bestowed an elaborate scientific theory to library classification introducing Analytico-Synthetic Classification system and devised Colon Classification. He has done a careful study of the 'Universe of Knowledge' and the various ways of the formation of new subjects. His studies have obviously to do a lot with basic ideas and their relations based on CoSSCo. Thus, there exist different classification schedules like enumerative scheme, almost enumerative scheme, almost faceted scheme, rigidly faceted scheme, analytico-synthetic classification (freely faceted) scheme etc. Of these approaches to classification, the present work deals with the last one [4].

The term 'Analytico Synthetic Classification' is a generic term to denote any scheme in which a compound subject is first analysed into its facets and synthesized in the verbal and notational plane. The analytico synthetic classification scheme includes two phases of work in three planes. Firstly, it analyses the subject into its facets in the idea plane. Secondly, it synthesizes in the verbal and notational plane. Ranganathan's principles of facet analysis are also well accepted today in connection with information storage and retrieval in even an online environment [5]. Being a freely faceted classification, it does not suggest any rigid facet formula but guides the

classifier to use postulates and principles while constructing a classification number. However, there have been attempts to make Colon Classification more and more analytico - synthetic. This tendency is increased gradually in different editions. The Colon Classification 7th edition is much more analytico-synthetic in nature than earlier editions.

3 Declarative Programming with Logical Approach

3.1 Logic based Representation of Knowledge

The logic-based representations of knowledge in developing AI based intelligent system have gained its popularity for various reasons. The usage of logic in different fields lead to the thinking of its application in the field of library and information science especially in thesaurus, indexing etc. And library classification is no exception. Example from Library Classification can be adopted here for easy demonstration. The five fundamental categories of the theory of classification, as generated by Prof Ranganathan for Colon Classification scheme, play an important role in classifying the thought-content of documents. Here is a fact *All diseases are belong to the matter property category of first round (IMP)* [6]. To develop the database, this fact can be expressed in logic formula as

$$\forall x. \text{Disease}(x) \rightarrow \text{MmatterProperty}(x),$$

and is read as 'For any category x in the world of medicine, if x is a disease, then x is matter property'. (However, to simplify the presentation for beginners the complication regarding rounds and levels is avoided in the above representation.) Thus, if the another fact '*Tuberculosis is a disease*' is added to the database in the form as

$$\forall x. \text{Tuberculosis}(x) \rightarrow \text{Disease}(x),$$

which is read as 'For any thing x in the world of medicine, if x is Tuberculosis, then x is a disease'. As these two statements are true then the following new statement must be true

$$\forall x. \text{Tuberculosis}(x) \rightarrow \text{MmatterProperty}(x),$$

that is 'Tuberculosis belongs to the matter property category'. Once the system reaches to this conclusion using the 'Rules of Inference'; it becomes the new added fact to the database and the system may proceed further to draw other conclusions to find out the indicator digit, isolate number and so on [7]. These approaches of presenting logical statements and their analysis can be introduced in developing an AI based system through suitable programming. Following sections will show the method of presenting logical facts for developing such system.

3.2 Propositional Logic

The simplest form of logic is the propositional logic or propositional calculus which is based on propositions (i.e. A properly formed statement) those have one of two different possible truth-values – either *true* or *false*. Following statements are typical examples of propositions -- ‘It’s raining’, ‘This rose is red’, and ‘Five plus ten equals thirteen’ or ‘Heart is a part of circulation system’. These statements are either true or false. However each of these statements is a proposition and cannot be broken down into its constituent parts assigning any truth-value. Because ‘It’s’, ‘This rose’, ‘Five plus’ or ‘Heart is’ can not be treated as a proposition as no truth-value can be assigned to them. The syntax of proposition leads to the simplest logic (i.e. the propositional logic) in which different statement looks something like “If Ranganathan is a man, then he is mortal.” Or “ If lung is a part of respiratory system, then it is an organ (implies it belongs to the category of Personality)”. The last part of these combined statements is based on Inference Rules [8]. An Inference Rule allows the deduction of a new statement from previously given statements; and the new statement is assured to be true if the original statements were true.

3.3 Predicate Logic

The predicate logic or predicate calculus is an extension of the notions of the propositional logic. For designing complicated AI based system, the above mentioned simple propositional logic is not very useful. Because the world of knowledge is not always expressible only by true or false propositions, rather in addition, is needed to be expressed in terms of objects and postulated relationships between these objects. And also these relationships can be generalized over the classes of objects. Here lies the usefulness of predicate calculus to represent statements about specific objects or individuals.

Statements about individuals, both by themselves and in relation to other individuals are called predicate. A predicate is applied to a specific number of arguments and has a value either false or true when individuals are used as arguments. The formula of predicate is written as a predicate symbol followed by some number of arguments inside parenthesis. A predicate in its general form may be written as

$$p(t_1, t_2, t_3, \dots, t_n).$$

Where p is the predicate symbol and $t_1, t_2, t_3, \dots, t_n$ are terms [9]. A term can be a constant (a string of lowercase letters or enclosed with single quote ‘.....’), a variable (started with an uppercase letter) or the application of a function. The application of a function may be written as a function symbol followed by a list of arguments whereas each arguments is itself a term.

However, a predicate can have more than one argument. For example following predicates are valid.

Gold is valuable	valuable(gold).
Knee is a part of Leg	part_of(knee, leg).
Medicine is basic subject of disease	is_basic_subject_of(medicine, disease).
In relation to CC7, L is the number for medicine	cc7(medicine, number, 'L').
etc.	

3.4 First Order Predicate Logic

As the predicate logic has appeared as very general, two additions – functions and unification, to the predicate logic form another logic type to make the predicate logic easier and understandable without extending the range of expressions. Thus it is called as First Order Predicate Logic or FOPL [10]. FOPL deals with the truth or falsehood of a statement and also helps to express objects and their relation with other objects. In addition, FOPL provides function to describe the relation of an object with other objects. For example from the previous section, ‘Knee is a part of Leg’ is represented as `cc7(knee, part_of, leg)` ; here though `part_of` establishes the relation between objects `knee` and `leg` but `cc7` is the function or the predicate which has three arguments. (Detail about such presentations is discussed in the later sections). For a given set of functions describing properties or relations, it can be ascertained whether a particular function is true or false. FOPL allows formulating rules about the properties or relation of objects [11]. This FOPL is used in handling AI problems as it supports knowledge representation in a better way.

3.4.1 Advantages of First Order Predicate Logic

Solving of the most AI problems are related to the techniques of natural language presentation and their proper analysis. The FOPL helps in natural language presentation and processing. Due the following advantages [12], FOPL is used in solving AI based problem:

a) **Efficient Reasoning:**

FOPL uses strong algorithms for solving a theorem than any other logic. Introducing resolution and unification algorithms can make improvements.

b) **Logic Programming:**

FOPL programming environments have richer and more developed implementations of reasoning algorithms and better engineering support facilities than environments for any other logic.

c) **Logic Grammars:**

FOPL programming based analysis and synthesis of natural language has several advantages over other formal approaches and offers several well-developed tools for building logic grammars.

Logical analysis means rewriting a sentence in such a way so as to bring out the logical form. The exact formalism is of lesser importance, provided it is used consistently. Several comprehensive logical systems have been proposed as frame works for the study of natural language with special emphasis on the study of its semantics.

4 Declarative Language in Logic Programming

Logic Programming is not, as one may expect, a very low level part of computer science, working with the logic gates. Logic programming is rather an attempt to use predicate logic as a basis of a programming language, leading to some very high level language. Logic programming can be defined broadly as the use of symbolic logic for explicit presentation of problems and their associated knowledge bases, together with the use of controlled logical inference for the effective of those problem. Logic Programming is useful for stating problems and it is useful for representing the pragmatic information for effective problem solving. The clausal form has a canonical form for the resolution principle, which has a complete and sound inference system consisting of one inference rule [13]. Since 1970s, the concept of logic as a uniform language for data programs, query views and integrity constraint has gained theoretical and practical potentiality. In case of databases, logic programs have been regarded as a generalization of relational databases.

The outputs of such programming are actually logical inferences presented in symbolic form. The mechanical applications of such logical inferences are used to solution of problems, which is also again presented in symbolic form. In this respect logic programming represents the contribution of logic to the practical problem solving technology needed by a variety of disciplines, of which computer assisted mathematics and AI are notable examples. Its application in the field of Library and Information Science has been started. It also draws upon developments within computer science concerned with placing computer system, programming languages and programming methodology upon a logical and cohesive foundation [14]. Logic programming has aimed to bring together and solve distinct aims in logic and computing: making logic more practical and making computing more logical.

Computer Programming Languages may be broadly categorized considering their procedural and declarative aspects. As the declarative aspects are to be exploited in the logic programming, so

Logic Programming languages are easily distinguishable from conventional programming languages like BASIC, COBOL, FORTRAN, Pascal, C etc. Prolog is considered as more useful for present case study because it attempts to incorporate logic in programming. In principle, the Prolog programmer is only supposed to specify ‘what’ (logic) to be done by the program without worrying about ‘how’ (control) the goal should be achieved.

4.1 Declarative Aspects of Prolog

PROLOG (PROgramming in LOGic) has both declarative and procedural aspects. However, it is considered mostly as a declarative language due to its inclination towards the same. Prolog also has procedural aspects, known as Impure Prolog, which in fact, often defeats Prolog’s declarative interpretation. Still, Prolog is considered as the first few steps forward for better logic programming with its declarative syntax.

In Prolog, programs are structured in terms of relations, which is quite distinct from traditional languages that use functions. The conventional idea of calling a function, passing some values to a function via arguments and also the functions resuming some values to the called functions are not provided in Prolog. But it has the idea about expressing relation between entities. This particular feature of Prolog is exploited for the AI programming. Role of variables in Prolog is different from that in other traditional languages [15].

In a real world situation, there would be a number of alternative solutions for a given problem and deterministic characteristics of procedural languages force one solution, as they follow a particular procedure or algorithm. But in case of a natural language, a sentence can mean many things in different contexts or within a given context. The non-deterministic approach of declarative language allows programmers to provide alternate solutions. And the *cut* ‘!’ operation is used to reduce the number of solutions. Prolog has non-deterministic characteristics since several elements can be in a particular relation to a given element [16].

Prolog has got the distinct advantages of rich capability of expressing symbols viz., words, parts-of-speech, structured objects (i.e. phrase structured trees), graphs etc. Prolog is a high level language in which a programmer can express operations as symbols represented by atoms and strings, numbers etc. and structures represented by list of terms, without actually creating these structure with pointers as in conventional languages. In Prolog, complex structures can be

expressed in terms of abstract patterns. Prolog accepts information at a very abstract level in terms of a set of facts and express complex retrieval operations via built-in inference mechanisms.

Prolog does not have iteration statements like *do* loops, *for* loops, *while* loops, assignment operator facilities, *go to* statements, *if-then-else* statements etc. which are very common for traditional languages [17]. Prolog provides recursion and not iteration as in conventional languages. In fact, recursion is the normal and natural way of viewing data structures and programs. Natural languages often use recursive structures like prepositional phrases, adjectival phrases, noun phrases etc. Hence the Prolog behaves in the better way in natural language processing.

5 Declarative Language for Natural Language Processing

Natural Language understanding requires knowledge of how the words are formed, how the word intern forms clauses and sentences. In a natural language, sentences are not just arbitrary sequences of words. A reasonable sentence cannot be formed bringing any set of words together. There is a syntax/grammar accepted for arranging words in a meaningful sequence. In the other words, a grammar for a language is a set of rules for specifying what sequences of words are acceptable as sentences in the framework of that language. It also indicates the way to group words together into phrases and the logical ordering of the phrases too [18]. In this regard, the use of logic to describe and to process natural language has gained importance with the emergence of logic programming.

The expressive titles of documents are prepared in natural language using abstract, titles and keywords. The notion of the present case study is to input such expressive titles to the system for processing. The system accepts and parses these natural language expressive titles to check their syntax with the help of lexicon and grammar formalism. These titles are analyzed in two phases. The first one, the syntactic analysis, would check the syntactical correctness of structure of sentences (i.e. expressive titles) as per the grammar formalism which defines the correct structure in the light of phrase structure tree depending on the lexical status of the words or word compounds [19]. A number of grammar formalisms have been designed as extension of logic programs. Many such logic grammars have been successfully implemented in Prolog. Prolog allows defining the logically correct general structure of a sentence in natural language that helps in determining the correctness of a specific sentence in future.

For example,

- s --> np, pp.
- np --> adj, noun.
- np --> noun.
- pp --> pre, np.

where **s** implies sentence, **np** implies noun phrase, **pp** implies prepositional phrase, **adj** implies adjectives and **det** implies determiner (i.e. a, an, the etc). A logically correct expressive title consists of noun phrase followed by a prepositional phrase; a noun phrase may consist of simple a noun or an adjective determiner followed by a noun; a prepositional phrase may consist of a preposition followed by a noun phrase.

A noun phrase is a self-contained unit and can fully interpreted by the time it has been read. That means it can always be determined what objects a noun phrase refers to. However, the important part is that the noun phrase is the valuable aid to achieve the objective of this present study. Because once these noun phrases are identified, then it is straightforward for the expert system to extract all information about it from the knowledge base. The second phase, the semantic analysis, would represent the meaning of the words in the specific subject domain i.e. the knowledge base developed in the light of Ranganathan's ideas regarding the analytico synthetic classification. Thus, all facets related to this domain are represented in terms of their fundamental categories, number etc.

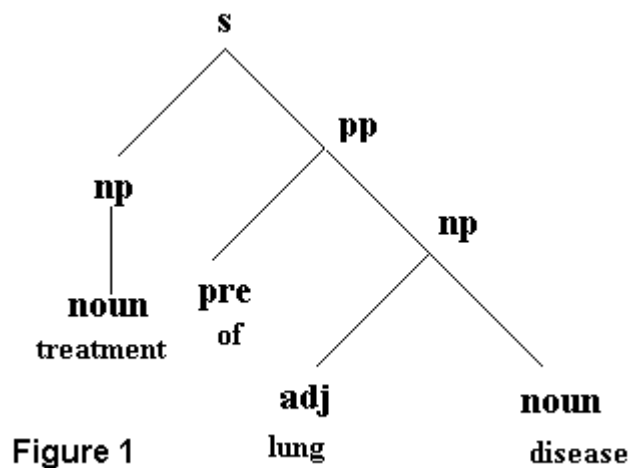


Figure 1

The above figure shows the phrase structure tree to understand the anatomy of a valid expressive title. Hence if the role of words in a sentence e.g. 'treatment of lung disease' be provided to the system as

- adj --> [lung].
- noun --> [disease].

noun -- > [treatment].

prep -- > [of].

Then the system will be able to check the sentence as a correct one [20]. And it will pick up the facets (i.e. noun phrase) like lung, disease, and treatment.

6 Knowledge Base for Library Classification

It is already mentioned that there exist CoSSCo relationships among all subject members of the Universe of Subjects; and for library classification whether these relationships can be represented by declarative aspects of programming language and can be handled by logic programming technique are main goals of this study. The basic principle of logic programming is to solve problems that involve objects and the relationship between those objects. So the subject members (Basic Facets and Isolate Facets) are theoretically considered as objects for the study. While entering into computer programming in the real world situation appears as a challenge. Prolog is chosen due to its some positive and special features mentioned earlier. Developing domain knowledge (or world of knowledge) using Prolog code in the computer system was a real problem. Here lies the reason of determining technique of knowledge representation, which is a basic requirement for any AI based system. In relation to the Colon Classification (7th edition) the domain knowledge like “Lung belongs to the category of Personality” or “Medicine is the basic subject of Lung” etc are represented as Prolog facts. Thus, while developing such system using Prolog, the following three major concepts [21] are to be considered carefully

- a) Declaring facts about objects and their relationships to represent domain knowledge,
- b) Defining rules to draw inferences using facts, and
- c) Asking questions about objects and their relationship.

6.1 Facts

A fact is a statement generated by heuristics from the real world situation. For writing facts in Prolog code, the following syntax is to be maintained [22]

- the names of all relationship and objects must begin with a lower case letter e.g. medicine, lung, basic_subject, cc7 etc. It is to be noted that as it is case sensitive, Lung is different from lung; Lung is a variable while lung is a constant.
- the relationship is written first, and objects are written separated by commas, and the objects are enclosed by a pair of round brackets i.e. starter and arvester.
e.g. basic_subject_of(medicine, lung).
 category_of(lung, personality).

isolate_number_of(lung, '45').

cc7(medicine, number, 'L').

- a period '.' (full stop) must appear to end the fact.

Facts are expressed in the form of predicate and its arguments. The objects that are enclosed within the round brackets are also known as arguments and the relationship is called predicate here. Facts are an essential part of prolog program. When a fact is included in a program, it is considered to have true value. If a fact is not included in a program, the fact is treated as false for the program. Following discussion will help to develop a knowledge base using facts for classification system.

In respect of CC7, if one wants to construct a classification number for a document on 'lung disease', two isolates viz. lung and disease are trivially identified. To construct classification number for lung the following five sets of information are essential [23].

- S1 Basic subject of lung is medicine.
- S2 The category of lung is personality.
- S3 Isolate number of lung is '45'.
- S4 Number of medicine is 'L'.
- S5 Indicator digit of personality is ','.

Similarly, for disease or any other isolate facet appeared in the expressive title, above mentioned five sets of respective information is to be incorporated in the system. However these statements if analysed carefully following three situations can be identified easily

- (i) The first three statements (S1 to S3) are purely 'lung' oriented.
- (ii) The fourth statement (S4) is related to the basic subject i.e. medicine and also related to the first statement (S1) as well.
- (iii) The last statement (S5) is related to the personality i.e. one of the five fundamental categories and also related to the second statement (S2) as well.

Frame Based Knowledge Representation is adopted for this study because it allows the hierarchical structure of the schedule. Object representation can be considered in the form of frame. These frames are identified with the characteristic objects of the domain. If any information is required about any one of these objects, the appropriate unit is accessed and all the relevant facts about the objects are retrieved at once. In general, an object and its characteristic objects may be represented as

(object, attribute-j, value-j) (j = 1, 2,, n)

The predicate is taken for this study as *cc7*, so the general form may be as

cc7(object, attribute-j, value-j). (j = 1, 2,, n)

where first argument is an objects, the second argument position is filled by the attributes of the object and the third argument is the respective values against respective attributes.

For example ‘*Lung belongs to the basic subject Medicine, its category is personality and its number is 45.*’ When this statement is represented as frame associated with the object ‘*lung*’

(object)	(attribute)	(value)
lung	basic subject	medicine
	category	personality
	number	'45'

Thus for a frame based knowledge representation model, a common structure be prepared instead of creating several independent frame [24]. For example, the S1 says ‘Basic subject of lung is Medicine’, where, *object* is lung, *attribute* is basic subject and the corresponding *value* is medicine. Thus it is represented as *cc7*(lung, basic_subject, medicine). to describe the form of different entities of the statement which is known as *term* in syntactic level. In the semantic level, it is considered that a meaning to each type of syntactic entity has been assigned and *cc7*(lung, basic_subject, medicine). is called a *fact*²⁵. Thus the above-mentioned five sets of information may be expressed as following Prolog facts

- F1 *cc7*(lung, basic_subject, medicine).
- F2 *cc7*(lung, category, personality).
- F3 *cc7*(lung, number, '45').
- F4 *cc7*(personality, symbol, ',').
- F5 *cc7*(medicine, number, 'L').

Obviously, facts F4 and F5 are to be mentioned once only for whole knowledge base. And apparently it seems three facts (same as F1, F2, F3) are required for each and every isolate ideas.

But according to the Medicine schedule of the Colon Classification 7th edition

- 4 Respiratory System
- 45 Lung

i.e. lung is a part of respiratory system. The basic subject and category of the respiratory system will be same for lung, i.e. attributes and values in the light of the facts regarding basic subject (F1) and category (F2) of the super ordinate ideas (basically topmost isolate idea) inherit to its subordinate ideas. So those facts can be declare once for the topmost isolate ideas and `part_of` relationship can be adopted to declare the fact like `cc7(lung, part_of, respiratory^system)` merging the facts F1 and F2 for the object lung. As per this representation, the topmost isolate of each chain will contain the attributes and respective values regarding its category, basic subject.

```
cc7(respiratory^system,  basic_subject,      medicine).
cc7(respiratory^system,  category,          personality).
cc7(respiratory^system,  number,            '4').

cc7(lung,                part_of,          respiratory^system).
cc7(lung,                number,           '45').
```

Thus, the number of facts can be reduced but the inheritance mechanism is adopted to make the system sufficiently knowledgeable with increasing efficiency [26].

6.2 Rules

A rule is a general statement about objects and their relationships. Rules are constructed on the basis of the logical analysis, heuristics, and conventions and also on the basis of some other relationships related to the objectives of the systems.

The general form of a rule is

$$a(\dots) \text{ :- } b(\dots), c(\dots), \dots \dots, z(\dots).$$

where `a`, `b`, `c`,, `z` are predicates and ... within parenthesis to represent arguments in any nature and any number. The structure of a rule consists of a head and a body; a connected symbol `:-` (a colon immediately followed by a hyphen) appears in between. Hence `a(...)` is the head of the rule and `b(...), c(...), ..., z(...)`. called the constraints/conditions and constitutes the body of the rule. The symbol `:-` is pronounced as 'if' and ','(comma) of the body is considered as 'and'. Hence the above rule may be described as the head `a(...)` is true if all the facts of the body `b(...), c(...), ..., z(...)` i.e. conjunction of goals must be satisfied one after the other. Ultimately a rule plays the role of a 'super fact' [27].

Now for constructing classification numbers, some rules are to be developed following the principles, postulates and canons of the theory of classification as developed by Dr S R Ranganathan. One such rule may be written in Prolog as

```
class_number_of(Term) :-  
    cc7(Term, basic_subject, Basic),  
    cc7(Basic, number, Basic_number),  
    cc7(Term, category, Category),  
    cc7(Category, symbol, Symbol),  
    cc7(Term, number, Number),  
    write(Basic_number),  
    write(Symbol),  
    write(Number).
```

When the system is asked to build Classification number for 'respiratory system' in Colon Classification, i.e. `class_number_of(respiratory^system)`. The output will be **L,4**. This is a very simple rule to build classification number basically for the topmost isolate ideas. But in the earlier section, it has already been mentioned that to develop a powerful knowledge base system, some complex concept like inheritance mechanism is to be adopted and the technique of knowledge representation are also changed accordingly using `part_of` inheritance. Demonstration in detail in this regard is included in the next section.

7 Knowledge Based Computer System : A Case Study for Colon Classification

A sample Knowledge Based Computer System is well demonstrated in three parts viz. knowledge base, programming, working procedure to get result and are included in Appendices I, II and III respectively. For these purposes Prolog language is chosen and Arity Prolog Compiler/Interpreter Version 6.1.28 is used.

The knowledge base is developed following the techniques discussed earlier for the Medicine schedule of Colon Classification 7th edition and stored as prolog facts in the *med.ari* file (Appendix I). The predicate name for such prolog facts is given as `cc7`, which consists of three arguments. At the beginning, indicator digits for five fundamental categories are declared along with the respective category name. Then a few facts for different categories like personality, matter-property, and energy are developed in such a way that the `part-of` inheritance mechanism could be exploited. For example, outer nose is a part of nose and nose is a part of respiratory system; so basic subject and category of respiratory system will be the same and will inherit to

nose and then to outer nose. For this `has_value` predicate gives the respective rule. All facts are developed in the same line of thinking but due to space limitations a few facts are shown only here for demonstration.

A small prolog program (*es.ari*) is developed to show the steps of constructing classification numbers (Appendix II). As this program is developed using the Arity Prolog Interpreter, some in-built predicates like `write`, `read_line`, `open`, `create`, `nl` etc are used, and detail discussion is avoided here. In addition some new predicates are added to construct rules as required following the theory of classification. The names of such predicates are so chosen that they may be self-explanatory one as far as practicable. The purposes of these predicates are also well documented within the program.

However, the predicate `build` is a very general rule meant to open an input filename in which expressive titles to be classified are stored and on completion of the process it stores the result in an output file. The `start` rule is used to read words of the titles as isolate terms with the help of `go` rule being supported by `class_number_of` rule. This `class_number_of` rule is again guided by `split_terms` rule, which helps in splitting compound concepts into individual isolates to pick up their indicator digits and isolate numbers through inheritance mechanism if required.

Lastly, the working procedure is shown for the title 'treatment of lung tumour' (Appendix III). After reconsultion of file *es.ari*, while the `build` is given against the '?' prompt, it asks to enter the input file name, *k1* is supplied as a file name in which the titles are stored. For example, only one title (i.e. 'treatment of lung tumour') is stored. Immediately after that it asks to enter the output file name, *k2* is entered as output file name in which the result is stored. Simultaneously the result i.e. classification number **L,45;472:6** is also displayed onto the screen.

8 Conclusion

The emergence of Artificial Intelligence has changed the direction of thinking in different areas in the field of Library and Information Science. With this approach automatization of library classification will be easier in the automatic environment. Though numerous attempts had been made to design powerful automatic classification systems but those could not give expected fruitful results. However, the main problem was the analysis of subject and finding out the subject propositions. Being a purely mental process, classification demanded human intelligence for analysing the title to find out its basic subject and other facets, if any, along with its category etc and synthesizing the facets to construct the classification number.

The present study with declarative language has paved the way towards developing AI based library system through logic programming techniques, as natural language processing could be possible through this approach. Prolog has proved its worthiness in this respect. Though Prolog has both the declarative and procedural aspects, its declarative aspect is used in AI based programming for natural language analysis. An exhausted study is carried out to exploit facilities of natural language processing with Prolog. But due to shortage of space, detail discussion and demonstration of natural language processing could not be included here in full. However, it is hoped that demonstration included in appendices (Appendix I, II & III) and discussed in the earlier section will help to have a clear idea about the new line of thinking towards the application of logic programming in the field of Library and Information Science.

9 References

- [1] A. R. D. Prasad. Logic and logic programming. In: A. R. D. Prasad (ed). *Artificial Intelligence and its application to Library and Information work. DRTC Refresher Workshop held at Documentation Research and Training Centre, Bangalore. 26-28 May, 1993.* (Documentation Research and Training Centre. Bangalore. 1993). Paper F. pp5-6.
- [2] Fernando C. N. Pereira and Stuart M Shieber. *Prolog and natural language processing.* (CLSI Lecture notes, no. 10). (Centre for the Study of Language and Information (CSLI), Standford, ©1987), pp 3 ISBN 0-937073-18-0
- [3] Seng Wai Loke & Andrew Davison. *Logic programming with the World Wide Web.* (<http://www.cs.unc.edu/~barman/HT96/P14/lpwww>, Last update June 13, 2002) (Visited on November 18, 2002)
- [4] FACETED CLASSIFICATION. Ranganathan : Ahead of his Century. (<http://www.slais.ubc.ca/courses/libr517/winter2000/Group7/facet.html>, Winter 2000) (Visited on October 12, 2002).
- [5] Aimee Glassel. *Was Ranganathan a Yahoo?.* End User's Corner. (<http://scout.cs.wisc.edu/toolkit/enduser/archive/1998/euc-9803.html>, 1998) (Visited on Augr 17, 2002)
- [6] S. R. Ranganathan. *Colon classification. 7th ed.* (Sarada Ranganathan Endowment for Library Science, Bangalore . 1989). p39-41. ISBN 81-85273-11-1
- [7] Avron Barr and Edward A Feigenbaum, (eds). *The hand book of artificial intelligence.* (Addison Wesley Publishing, Massachusetts, 1986.) pp154-155. ISBN 0-201-11811-4

- [8] P. K. Panigrahi. *Design and development of a knowledge representation model for analytico synthetic classification*. (Unpublished Ph D Thesis. Vidyasagar University, Medinipur, India, 1996). pp60-61.
- [9] John Malps. *Prolog : a relational language and its applications*. (Prentice Hall, New Jersey, 1987), pp26-27. ISBN 0-13-730805-1 025
- [10] Avron Barr and Edward A Feigenbaum, (eds). *The hand book of artificial intelligence*. (Addison Wesley Publishing, Massachusetts, 1986.) pp165. ISBN 0-201-11811-4
- [11] FIRST ORDER Predicate Logic. (http://www.cs.umu.se/~hegner/courses/TDBB08/v98b/slides4/fo1_4.pdf) (Visited on Nov 18, 2002).
- [12] A. R. D. Prasad. *Application of computer based natural language processing tools and techniques in developing subject indexing language*. (Unpublished Ph D Thesis. Karnataka University, Dharwad, India. 1993.). pp135-136.
- [13] R. A. Kowalski and C. J. Hogger. Logic programming. In: Stuart C. Shapiro (ed). *Encyclopaedia of Artificial Intelligence*. Vol 1. (John Wiley and Sons, New York , 1987), pp544-558 ISBN 0-471-62974-x
- [14] Terry Wingrad. Beyond programming languages. *Communications of the A C M*. 22(7), 1979. pp391-401.
- [15] WHAT ARE Logic Programming and Prolog. (<http://www.biocheminf.com/moirai/Cs-magenta/prolog.html>, Last update Mar 5, 2002) (Visited on Nov 11, 2002).
- [16] W. P. Clocksin and C. S. Mellish. *Programming in PROLOG. 3rd rev ext ed.* (Narosa, New Delhi, 1993), pp 75-76. ISBN 81-85198-15-2
- [17] Patrick Saint-Dizier. *An introduction to programming in PROLOG*. (Springer Verlag, New York, 1990), p1-2. ISBN 0-387-97144-0
- [18] P. K. Panigrahi. An Artificial Intelligence approach towards Automatic Classification(Part 1). *IASLIC Bulletin*. 45(2), June 2000, pp 73-81.
- [19] Noam Chomsky. *Aspect of the theory syntax*. (Massachusetts, The MIT Press, 1965), pp63-106. ISBN 262 53007 4 (pbk)
- [20] P. K. Panigrahi. An Artificial Intelligence approach towards Automatic Classification(Part 2). *IASLIC Bulletin*. 45(3), September 2000, pp 105-118.
- [21] W. P. Clocksin and C. S. Mellish. *Programming in PROLOG. 3rd rev ext ed.* (Narosa, New Delhi. 1993), pp 2. ISBN 81-85198-15-2
- [22] Patrick Saint-Dizier. *An introduction to programming in PROLOG*. (Springer Verlag, New York, 1990), pp3-9. ISBN 0-387-97144-0
- [23] S R Ranganathan. *Colon classification. 7th ed.* (Sarada Ranganathan Endowment for Library Science, Bangalore . 1989). pp 229-234. ISBN 81-85273-11-1
- [24] P. K. Panigrahi. *Design and development of a knowledge representation model for analytico synthetic classification*. (Unpublished Ph D Thesis. Vidyasagar University, Medinipur , 1996). pp.87-89

- [25] Patrick Saint-Dizier. *An introduction to programming in PROLOG*. (Springer Verlag, New York, 1990), pp20-23. ISBN 0-387-97144-0
- [26] P. K. Panigrahi. *Design and development of a knowledge representation model for analytico synthetic classification*. (Unpublished Ph D Thesis. Vidyasagar University, Medinipur , 1996). pp.95-100
- [27] Patrick Saint-Dizier. *An introduction to programming in PROLOG*. (Springer Verlag, New York, 1990), pp20-23. ISBN 0-387-97144-0

Appendix I

A sample knowledge base to represent the medicine schedule as a collection of facts (file name - med.ari)

% Indicator Digits

cc7(medicine,	number,	'L').
cc7(personality,	symbol,	',').
cc7(matter_property,	symbol,	';').
cc7(energy,	symbol,	':').
cc7(space,	symbol,	',').
cc7(time,	symbol,	'“').

% Personality Schedule

cc7(respiratory^system,	category,	personality).
cc7(respiratory^system,	number,	'4').
cc7(respiratory^system,	basic_subject,	medicine).

cc7(nose,	part_of,	respiratory^system).
cc7(nose,	number,	'41').
cc7(outer^nose,	part_of,	nose).
cc7(outer^nose,	number,	'411').
cc7(nasal,	part_of,	nose).
cc7(nasal,	number,	'412').
cc7(larynx,	part_of,	respiratory^system).
cc7(larynx,	number,	'42').
cc7(trachea,	part_of,	respiratory^system).
cc7(trachea,	number,	'43').
cc7(bronchi,	part_of,	respiratory^system).
cc7(bronchi,	number,	'44').
cc7(lung,	part_of,	respiratory^system).
cc7(lung,	number,	'45').

% Matter Property Schedule

cc7(disease,	category,	matter_property).
cc7(disease,	number,	'4').
cc7(disease,	basic_subject,	medicine).
cc7(structural^disease,	part_of,	disease).

```
cc7(structural^disease,      number,      '47').
cc7(tumour,                  part_of,     structural^disease).
cc7(tumour,                  number,      '472').
%Energy Schedule
cc7(treatment,              category,    energy).
cc7(treatment,              number,      '6').
cc7(treatment,              basic_subject, medicine).
```

Appendix II

Some sample rules to show a simple method of building classification number (file name – es.ari)

```
:-reconsult(med).
```

```
build:- write('Enter Input File Name: '),
        read_line(0, Infile),
        open(Input, Infile, r),
        recorda(file,Input,_),
        write('Enter Output File Name: '),
        read_line(0, Outfile),
        create(Output, Outfile),
        recorda(out, Output,_),
        start(Input,Output),!.
build.
```

```
start(Input,Output):-
    read(Input,Words),!,
    write(Words),nl,
    write(Output, Words),nl(Output),
    write('WORDS: '), write(Words),nl,
    write(Output,'WORDS: '), write(Output,Words),
    go(Words),
    nl(Output),nl(Output),
    fail.
```

```
start(,_):-
    close(Input),
    close(Output).
```

```
go(Term):- class_number_of(Term),!.
```

```
class_number_of(Term):- write('L'), split_terms(Term).
```

%Splitting into individual Isolates from Compound Terms

```
split_terms(Term1^Term2^Term3):-    class_number(Term1),
                                     split_terms(Term2^Term3).
split_terms(Term1^Term2):-          class_number(Term1), class_number(Term2).
split_terms(Term1):-                class_number(Term1).

class_number(Term):- find_category(Term, Basic_No, Category, Symbol, Number),
                    %    write(Basic_No),
                    write(Symbol), write(Number).
```

% To perform 'Inheritance Mechanism' to pick up basic subject, categories, % indicator digits of isolates

```
has_value(Object, Slot, V):-        cc7(Object, Slot, V),!.
has_value(Object, Slot, V):-        cc7(Object, part_of, Superconcept),
                                     has_value(Superconcept, Slot, V).
```

% To find the Basic Subject, Numbers; Categories of different Isolate terms, % their Symbol, Numbers etc

```
find_category(Term, Basic_No, Category, Symbol, Number):-
    has_value(Term,    basic_subject,    Basic),
    has_value(Basic,  number,            Basic_No),
    has_value(Term,   category,          Category),
    cc7(Category,    symbol,             Symbol),
    has_value(Term,  number,             Number).
```

Appendix III

Following is the screen layout of Arity Prolog compiler for building the classification number of 'treatment of lung tumour'.

```
?:-reconsult(es).

yes
?-build.
Enter Input File Name : k1
Enter Output File Name : k2
Lung ^ tumour ^ treatment
WORDS : lung ^ tumour ^ treatment
L,45;472:6

yes
?- _
```

Acknowledgements

I express deep gratitude to Prof Asok Basu, University Librarian (Retd) & Guest Teacher, Dept. of Library and Information Science, University of Calcutta, Kolkata and Prof A R D Prasad, Documentation Research and Training Centre, Indian Statistical Institute, Bangalore for their valuable guidance and suggestion in relation to this research study.

Author's Detail :

Dr P K Panigrahi obtained M Sc (Applied Mathematics and Computer Programming) and BLISc(Gold Medalist) degree from Vidysagar University and ADISc (Associateship in Documentation and Information Science) from Documentation Research and Training Centre, Indian Statistical Institute, Bangalore, India. He has been awarded Ph D degree by Vidyasagar University. He is engaged to perform research in Library Automation in India. Dozens of his research papers are published in national and international level journals. He is teaching IT applications in library science since 1992 being attached with Vidasagar University, Kalyani University and Calcutta University in India.