LABEO: A KNOWLEDGE-BASED EXPERT SYSTEM

FOR THE "ANIMATION" OF LEGAL TEXTS

Costantino CIAMPI

1. TRENDS IN LEGAL INFORMATION RESEARCH AND RETRIEVAL: FROM LEGAL DOCUMENTATION RETRIEVAL SYSTEMS TO KNOWLEDGE-BASED SYSTEMS

The history of the application of the computer and information science techniques to legal documentation goes back almost thirty years, during which time there has been an enormous progress in hardware technology and a slower, more uncertain evolution in software and in the methodologies related to documentation analysis.

Analytical documentation methodologies have, in particular, remained almost unchanged for more than a decade. This means that, in the great majority of documentation retrieval systems currently in operation, the legal texts which are still being fed in will then be retrieved in a relatively banal fashion by simply making a comparison (between the words in the query and the words in the text) at the level of lexical form (whole or truncated), of lemmata or, even more so, at the level of some embryonic syntactic-semantic analysis.

The inadequacies of these methods have, to some extent, been masked so far by the ever-increasing operational speed of modern computers and by their greater interaction with the user, a characteristic feature of recent system architecture.

However, the first part of this article is critical of some of the aspects of these traditional methods, thereby justifying the change in direction offered by knowledge-based systems, and by one of these in particular, the LABEO project, which will be described in detail in the second part of this article.

1.1. The main features from a linguistic point of view of present documentation systems can be summarized as follows:

- the word is the distinguishable unit (in inverted files);

- these units are classifiable in stop words (or functional words) and full words according to their grammatical category and their frequency in the language;

- each word (inflected or conjugated) can be reduced to a basic form (lemma);

- there is a co-occurrence or proximity relation between these units which determines their syntactic/semantic relation within the documentation unit that has been decided upon in a conventional way for practical purposes (for example, case law abstracts or sections of an Act).

Although the limitations involved in these fundamental choices have been adequately analysed in the technical literature, it will be useful here to bring them to mind once again:

- a) the mistaken idea that the meaning of the sentence is located in the subject/predicate pair is re-enforced, whereas it is often the adverbial occurrences, situations, circumstances, which characterize the sentence;

- b) the morphemes revealing the sequence of time or the mode of action are neglected;

- c) no relation either of coordination or subordination is thought to exist between two propositions.

1.2. From the information science viewpoint, these present systems are generally based on two files: the text file and the index file.

The former simply contains the complete text of the stored document (or else suitable "substitutions" for the original texts: summaries, abstracts, epitomes, regests and other means for the conceptual reduction of the document whilst still properly representing its content).

The latter is an alphabetical list of all the words to be found in the (original or substitute) texts, excluding grammatical words (such as articles, prepositions, conjunctions and pronouns) which have been defined as "empty" of any independent meaning. The positions or "addresses" of the words appearing within the

analysed texts follow the alphabetically listed words. This "address" is expressed in a structured code which identifies the document, the paragraph, the sentence and generally even the place occupied by the word in the text. The "addresses" in the index-file serve as control words for gaining access to the text file.

Whoever is searching for information (the user of the system) must analyse the query he intends to put to the computer, dividing it into concepts, translating those concepts into a list of research terms, linking these terms with the boolean operators OR, AND, NOT, etc., and then wait for the computer's reply. The index-file from the computer's point of view is a table (inverted file) or matrix which permits the information to be retrieved more rapidly.

Naturally, I have, for the sake of brevity, represented traditional documentation retrieval systems in a very simplified way.

In reality, there are an infinity of variations. Systems have, for example, been adopted in which indexing is done by hand rather than being electronically produced as a lexical sequence, by forms or lemmata, of all the words appearing in the stored text. In these systems an expert indexes the content of the texts choosing key words, descriptors, terms and classification codes which, when ordered alphabetically, constitute the headings of the index-file for automated consultation.

Experts in documentation and information science have compared the advantages and disadvantages of these two systems (the free text and the key-word systems). It was quickly understood that, in fact, all data filing systems wherein there is no keyword indexing or any other form of classification are only able to provide the researcher with information about the "occurrence" of the requested word in the stored text. It should be noted that whilst this may satisfy the philologist, the legal lexicographer or the legislator who is careful about the use of words - and therefore interested in learning about all the contexts in which a specific word he is on the on point of using again or varying has been used - it may not satisfy the legal operator who needs to retrieve all the documents relating to a particular legal problem or concept no matter what the verbal expression has been used by the document's author (legislator, judge, legal academic, etc.).

1.3. To overcome "silence" in the system" (that is, where the documentation system states that the requested word does not appear in any of the stored documents), technology is required

which will result in the automated expansion of the terms provided by the user as retrieval keys to all the possible morphologic, syntactic, and semantic variations of those expressions.

This means that the lawyers and linguists need to pre-establish a correspondence table (to be continuously up-dated) between the words in the user's language and the words which can be linked to them as synonyms, equivalents or in relations of correlation from genus to species and vice versa, or of generic association. Although the development of such linguistic tools (thesauri, grammatical generators and so on) has found fertile ground in the more recent documentation systems, the problem at a conceptual level had already been posed in the early systems and the need to solve it has been strongly felt.

Thesauri, in particular, face not only the usual linguistic problems (declination and conjugation of nouns, adjectives and verbs, polysemy and synonymity) but also and above all the problems arising out of the conceptual organization of the legal discourse.

Preparing thesauri with the assistance of the computer and their "on line" utilization were the first challenge taken up by advanced documentation systems. This meant that conceptual research became possible which was wider and more satisfactory than simply searching the documents by comparing the user's words with those contained in the text of the analysed documents.

A second and more difficult challenge faced by advanced documentation systems was in creating a system which extrapolates by itself the useful information contained in the document, relieving the user of this task, thereby creating a system of automated consultation comparable to the results reached in gaining immediate access to "useful" information by consulting a human expert. Once these systems have been implemented, I do not know whether it would still be correct to call them documentation systems (even if they are given the additional qualification of "advanced") or whether they form a category of their own which is destined to bring some other basic categories, such as those relating to knowledge, its production, conservation and transmission, into discussion.

I shall attempt, in the following section, to analyse more closely the terms of this challenge.

1.4. Generally, the user of traditional legal documentation systems is provided with the documents containing information

relevant for solving the proposed legal problem; and then it is the user and the interested party who must read, interpret and reason through these documents to arrive at the solution to the problem, facing the obscurity of legal texts, the sometimes contorted style of our legislators or legal writers, the difficulties created by the "allusiveness" of legal language and the cross-referencing from one text to another, both the burden and the joy of our legal interpreters.

Knowledge-based systems, on the contrary, tend to free the researcher from these heavy and dubious tasks, providing solutions directed at the specific problems posed. They give neither the text of the document which needs reading nor references to publications which must be found and then interpreted; but they tell the researcher precisely whether he has the right, the faculty or the power to carry out a certain activity, how he must carry it out and legal consequence which may flow from his failure to comply with the rules.

If documentation systems tend to substitute libraries and documentation centres, knowledge-based systems represent a modern way of organizing and managing consultation by experts in individual disciplines and are, therefore, useful to legal operators, particularly in those areas wherein typical situations occur or wherein numerous problems of a similar kind arise: for example, in tax law, in that relating to succession, landlord and tenant law, to labour law and other similar areas.

Even the Italian Building Indemnity Law ("Condono Edilizio")(Law No. 47 of 28 February 1985) - chosen for the purposes of this article for testing the main operational functions of the LABEO system - seems to be an important example of a normative text worthy of becoming the subject of a knowledge-based system because of the obscure and contorted language in which it has been written and because of the gaps, redundancies and contradictions which, even after the first analysis, it has proved to be full of.

What is needed for constructing a knowledge-based "expert" system?

I shall limit myself here to noting that for building an expert system "cognitive acquisitions" or "structured knowledge" of a specific area need to be gathered and then fed into the computer along with the logical rules required for processing this knowledge.

The knowledge bases store something more than simple data in their memory.

345

Generally speaking, their patrimony equals the information available to experts and the rules regulating the use of this information for problem solving, for reasoning, and for decision-making.

The computer in knowledge-based systems is not merely used as an extremely efficient and fast machine for reading and printing the documents statically stored in its memory, but is utilized mainly as a logic machine for making more or less complicated calculation about the knowledge in a specific area of a discipline communicated to the computer in formal language.

The information, therefore, does not become, generally or solely, something to be "conserved" and "retrieved" ("called up") as in traditional documentation systems, but is also the subject of different logical operations aimed at transforming it into "new" information.

An auomated system which organizes and processes a set of legal knowledge in such a way that, when a query is put to it, it gives the user as a response the deontic characterization of the action he is about to perform and direct guide to the way he should behave in the future must store a catalogue of abstract legal fact situations with the related legal effects connected to them, a catalogue of questions to ask the user for gathering information on the concrete fact situation, and all the logical functions required for subsuming the concrete fact situation in the standard abstract scheme and for carrying out other formal logical calculations.

An "expert" system of such an evolved kind does not as yet exist although there has been much progress since this particular challenge was made.

Numerous experimental systems created at research institutions mainly in the United States of America and in Northern Europe are in existence today. These include those created in the United States at the University of Michigan, MIT of Boston, the American Bar Foundation of Illinois, Rutgers University and the Rand Corporation Laboratories in California, and in Northern Europe at the London School of Economics and the University of Stockholm.

The Istituto per la documentazione giuridica of the Italian National Research Council in Florence has fulfilled an important role in promoting research in this field in Italy, but also has had a strong influence on the international scene. Merit goes to it above all for testing (at the risk of sometimes being accused

of methodological eclecticism) the theories and techniques proposed by various different scholars (M. Sànchez-Mazas, L.E. Allen, C.E. Alchourròn and E. Bulygin, etc.).

Even the LABEO project - which has been chosen for testing the "normalization" of a normative corpus by the author of this article - whilst it has been developed with the contribution of researchers working outside the Istituto per la documentazione giuridica, has been influenced by precisely that eclecticism characterizing the research projects developed within the Institute.

Let me now discuss in some detail this research project - even though the achievement of all the aims of the LABEO project still seems somewhat far off - as the developments it appears to lead to are interesting if they are properly aimed at at integrating normalization techniques as well as at advanced automated documentation technology.

2. THE LABEO PROJECT: BASIC METHODOLOGICAL CHOICES, FUNCTIONS ALREADY IMPLEMENTED AND PROGRAMS IN THE COURSE OF DEVELOPMENT.

2.1. Introduction.

The LABEO Projects gets its name from the ancient Roman jurisconsult Marcus Antistius Labeo who lived in the time of Augustus, was a follower of stoic philosophy and a supporter of the introduction of formal logic into the analysis of legal reasoning.

The project, as has already been mentioned, has as its operational aim the preparation of a set of programs for the "animation" of legal texts (laws, decisions, doctrinal opinions, contracts, etc.) in the way described in the preceding paragraphs.

The system is based fundamentally on the decomposition of the text into constituent sentences, classified in antecedents and consequents; on the formal exercise of their syntactic and logical relations (interpretative logical formulae); on a relational management system of the constituent sentence base and of the formulae (suitably integrated by a thesaurus of descriptors relating to the sentences introduced into the knowledge base); on programs for manipulating the sentences, formulae descriptors and texts, with various purposes in mind (such as, for example, the normalized drafting of a text, the diagrammatic representation of it, the transformation of the

formulae, etc.); and, finally, on the functions of the interface between the system and the user for managing their dialogue and the operations involved in the consultation of the knowledge base.

After the project is completed, when LABEO is queried about a specific legal problem it will reply to the person making the query like the ancient Roman jurisconsult asking if certain conditions which are the logical and legal premise of the facts connected with the problem posed are present. Therefore, LABEO by relying on the user's replies and "his" reasoning rules, will deduce and express the legal consequences of the case.

2.2. Guiding Principles.

The following two fundamental directional paths were taken in planning LABEO:

a) the normalization technique, as proposed by the American academic Layman E. Allen, has been used for analysing and formalizing legal texts;

b) symbolic logic has been as the language for representing the legal content.

2.2.a. The Normalization Technique.

The normalization technique was created as a tool for improving the drafting of legal texts by eliminating any syntactic ambiguity from them.

By following the distinctions made by Allen, we can say that there is an imprecision in the meaning of a sentence (or of a text) when more than one meaning can be given to it.

The imprecision can be characterized as uncertainly or incompleteness depending on the expressions actually used (required for forming the sentence) or on the fact that some expressions have been omitted.

Uncertainty can be characterized further in syntactic or semantic uncertainty depending on whether it involves the syntactic or the semantic aspects of the sentence. The syntactic aspects of the sentence regard the way in which the semantic words (which delineate the objects of the universe of the discourse and their properties or relations) are combined within the structure of the sentence, through the expressions which express the syntactic relationships.

Syntactic uncertainty causes ambiguity in the sentence (syntactic ambiguity) to which several different alternative syntactic structures can be given.

348

The semantic aspects, instead, involve the semantic words. Semantic uncertainty can take one of two forms, either indeterminateness (or vagueness) or ambiguity. Semantic indeterminateness exists when we are unable to determine the width of a term exactly, that is, when we cannot determine as far as the meaning of a term is concerned, the boundary between that to which it refers and that to which it does not refer. There is, instead, semantic ambiguity when the term has several different meanings.

The normalization technique means producing a legal text which has no syntactic ambiguities.

It is well known that legal language (both that of the legislator and that found in legal dogma) is not a formalized language.

It, in fact, borrows the lexicon, the syntax and the meanings of the terms it utilizes from ordinary language. Like all specialized languages, although legal language recognizes the potentialities of technicalization or, in other words, of using technical words (or expressions) or technicalized words (or expressions), this technicalization is usually restricted to semantic terms and does not extend to terms expressing syntactic relationships.

Instead for expressing syntactic structures it totally relies on ordinary language. Not even the phenomenon of the technicalization of syntactic expressions which characterize scientific language where natural language expressions and the logical expressions in the same context are frequently used tending to interpret the former as synonyms of the latter occurs in legal language.

One of the features of legal language is, therefore, its considerable syntactic imprecision which, even if it rarely causes serious problems of interpretation, makes it considerably difficult to understand the text, hindering the lawyer in concentrating on questions of substantive interpretation (which usually concern the vagueness of the semantic terms). If the vagueness (which is to some degree the inevitable result of generality) of the expressions used by the legislator is capable of meeting important legal and social needs (to avoid an excessive fragmentation of cases in the discipline, to facilitate the adaptation of the case law to the particular fact situations and to changes in social conditions, etc.) and, therefore, be consciously introduced by the legislator, this is not usually so for ambiguities. It is very difficult to imagine syntactic ambiguity being deliberately introduced into a legislative text.

As a remedy for this kind of ambiguity, Allen proposes to apply the definition technique even in syntactic expressions and, specifically, as we shall see later in this paper, to introduce logic connectives and other connectives defined in terms of logic connectives into legal language. The definition of syntactic

expressions is particularly interesting as it has none of the restrictions of the definitions of semantic terms. It is, in fact, often possible to define a syntactic expression not only by other syntactic expressions (which must, in turn, be defined) but also by specifying the conditions whereby the proposition constructed through this expression turns out to be true or false (or alternatively, by specifying the condition from which the proposition can be deduced or which condition can be deduced from it).

The ambiguity of the syntactic structure may, however, involve not only the meaning of the syntactic terms but also determine their arguments or, in other words, the "field" to which the syntactic expression refers.

In order to eliminate this second kind of ambiguity, Allen proposes several graphic techniques permitting the unequivocal specification of that to which each syntactic expression refers performing, therefore, a function analogous to that of brackets in logical or mathematical expressions.

We will call "normalization" that procedure which leads to a reformulation of the legal text without any ambiguity ("normalized version").

The normalized version (on the level of propositional logic) of a legislative text is, in fact, the result of a sequence of "normalized provisions". Each sentence constituting a non atomic normalized provision (sentences which may be either atomic or molecular) is given a letter or a number, logical connectives express the connection between these sentences and the arguments related to those connectives can be identified on the basis of drafting rules. The atomic sentences are given a tag which unequivocally identifies them.

There is a great difference between the use of a formalized language (even if only partially as in our case) directly by someone drafting a normative act and his use of it to reformulate that which has been expressed in natural language in a earlier .normative act usually derived from some official document. Whilst in the former case some of the uncertainties which arise in the interpretation and enforcement of the law can effectively be overcome, in the latter, instead, the generating of a formalized text corresponding to a given document written in natural language does nothing more than suggest a particular interpretation of that document (or, if necessary, several alternative interpretations). The use of a formalized language could, nevertheless, allow us to formulate the interpretation or interpretations under consideration with more precision.

It is, in conclusion, to be noted that the normalization technique, as perfected by Allen so far and as applied in the LABEO system, concerns only the propositional structure of the text. The formalization is, therefore, limited to the relationships between propositions (and, to be more presise, only

to the relationships between propositions which can be expressed through true-functional connectives).

This means that the syntactic ambiguity within the sentences cannot be made unambiguous in the normalized text if the relationship between the expressions making up the sentence cannot be transformed into the relationship between separate sentences. This often cannot be done without heavily weighing down the text. It is, furthermore, impossible to adequately express the fact that several sentences refer to the same object without repeating the name of that object in each sentence (likewise for all the implicit references in the sentence under consideration which are only expressed in the adjoining sentences). A suitable formalism for the legal discourse must, instead, go as far as the level of predicate logic. That is it must, on the other hand, represent a natural development in the application of the normalization technique to the propositional structure.

2.2.b. <u>Symbolic Logic as a Language for Expressing Legal Content</u>.

The normalization technique produces a text in which part of the syntactic structure is expressed in logical language.

Although this leads to changes with respect to a natural language text, they are not as radical as may be believed. The normalization of a legal text, in particular, does not lead to every word in the normalized text being given a precise meaning. The formalization is, in fact, restricted to the syntactic structure taken into consideration but does not, if only marginally, involve the problems of ambiguity and semantic vagueness.

When uniformity in using the non formalized expressions is combined with the formalization of the syntactic terms, enough conditions are created for applying logical calculus. As far as the non formalized expressions and the semantic expressions, in particular, are concerned, all that is required is that they always be given the same meaning within the normalized text.

Utilizing logical methods for evaluating whether an inference is correct or for (automatically) developing an inference starting off from given premises assumes, in fact, that only the logical structure of the propositions occurring within the inference to be evaluated or in the premises from which to move is expressed in the language of formal logic. And the concept of the logical structure relates to the level to which the logical analysis is taken: the deeper the logical analysis of the inference desired and/or the greater the capacity to make inferences the greater the number of expressions to be translated into the formal language.

Likewise, for the application of the propositional calculus, a formalization of the expressions (such as "and", "not", "or",

"if", "unless", etc.) is sufficient. We believe we can interpret these as true-functional connectives between propositions and the modalities through which the arguments relating to those connectives are identified. For the purpose of applying the calculus of predicates, the formalization must also be extended to expressions (such as "all", "some", "few", etc.) which can be translated into quantifiers and to the modalities through which the relationship between a predicate and the terms to which it refers is expressed.

The formalization, for the application of modal logic, must also deal with the expressions ("it is possible that", "necessary that", etc.) that can be translated by the modal operators so that deontic logic can be applied to expressions that can be translated by deontic operators, and so forth.

The use of logically defined expressions in legal contexts has two strictly connected aspects:

1) The elimination of some of the inaccuracies in the meaning of the original text (those arising out of the ambiguity of the expressions substituted by the logical expressions);

2) The possibility of utilizing logical tools for evaluating and/or making inferences.

The choice of logical language for expressing legal content, a choice which is implicit in adopting the normalization technique, is justified as compared with other formal languages, by both the above-mentioned aspects for the following reasons:

- logic is closer to natural language and, in particular, to legal language and, therefore, its use facilitates the passage from natural to the formalized language;

- logic is more easily comprehensible as compared to graphic or net-type languages (for example, flow charts) which, even if they are able to effectively represent very simple contexts, are totally unsuitable for representing more complex contexts,

- it is mainly a declarative language lending itself with spontaneity to be used for representing facts, objects and relationships without a sequential procedure which reproduces the computer's functions (as occurs in "procedural" languages);

- it is the language which is part of our cultural tradition, already well tried in many different fields of science and for the most varying subjects, and is, therefore, particularly suitable for the law which touches on the most diverse sectors of society;

- it is the closer to recent trends in computer technology and, in particular, to expert systems than to

traditional programming (where it may be sufficient to interpret a computer program as the description of a sequence of actions).

The fact that logical tools can be used for evaluating or making inference imposes conditions for introducing computer technology. There are, obviously, limits to automating inferential operations but this does not affect the validity of that which can be created. It should be pointed out here that, in recent years, advances in computer science have produced languages and programming technology which are increasingly approaching logic. If, on the one hand, the use of logical language for expressing legal content has an element of continuity with tradition undoubtedly making its use in legal contexts less traumatic than the use of other formal languages, it represents, on the other, a link between the most recent research in computer technology and imposes conditions for an efficient utilization of knowledge in automated inference systems.

A final observation: a preliminary question with respect to those already considered is related to the actual applicability of logic to the law.

2.3. The present stage of the implementation of the Labeo system (version 1.1., updated to 31 march 1988).

The LABEO system is currently divided into 4 modules, integrated in a supervisory program (see Fig. 1):

1) a module for the analysis and drafting of legal texts and for the planning and management of the knowledge base;

2) a module for searching the system which utilizes the knowledge base and the information about the case provided by the user for extracting, through an inference engine, the relevant consequences relating to the question posed;

3) a module for drafting and managing thesauri which the user can refer to for assistance in formulating his query;

4) a module for storing, searching and updating the sentences and the interpretative formulas as well as the original texts and normalized results of the application of module No. 1.

Module No. 1: Normalization of the Texts and Preparation of the Production Rules of the Knowledge Base.

"Normalization" is a procedure which leads to the reformulation of a legal text without any syntactic ambiguity (normalized

version) (see example in the <u>Appendix</u>).

The procedure develops in several phases and requires the following input: 1) the identification of the minimum unities making up the original version of the text; 2) the description of the logical structure of the text in well formed formulas (WFF), which are called the interpretative formulae. At present the system can only treat propositional logic, although predicate and deontic logic are being implemented. For this reason the minimum unities are atomic statements, which we call elementary statements, and the interpretative formulas are the well formed formulas (WFF) of propositional calculus. The constituent sentences (each one represented by the label distinguishing it) in these formulas are correlated by the logic connectives of LABEO's language (AND, NOT, OR, IF-THEN, UNLESS, BUT OTHERWISE, etc.).

This module of the LABEO system provides us with the possibilities of: a) automatically drafting several normalized versions which are logically equivalent to the interpretative formula of the text originally furnished by the user but whose logical structures gradually become simplier (i. "complex"; ii. "plain"; iii. "basic"; iv. "elementary"); b) automatically translating the "elementary" interpretative formulas into rules for producing the legal effects expressed in the syntax required by LABEO's inference engine (the syntax is close to the formalism of the Horn clauses based on PROLOG).

<u>Module No. 2</u>: Consultation of the Knowledge Base.

This module mainly works with three components:

a) the long term memory which is made up of the rules for producing the legal effects arising out of the application of Module No. 1;

b) a very general algorithm, the inference engine which is substantially the implementation of the method of reasoning;

c) the short term memory constituted by the information, that is, by the facts of the individual case supplied by the user in reply to the questions put by the system.

By activating this module the user is presented with all the possible queries to be put to the system. These are all the consequences of the hypothetical elementary norms represented in LABEO's knowledge base.

The user is therefore called upon to specify the "legal effect" he is interested in. The computer asks whether or not facts exist which condition the legal effect dealt with (those which are a sufficient condition of it). If those facts are, in turn, legal effects (the consequence which a rule of the system links to other facts), the system indicates this. In this case, the user may not only state that that particular element of the case has

or has not occurred (answering yes or no respectively to the query) but he may also choose the P option. In the latter hypothesis, the system moves on automatically to consider the conditions to which the element of the case is linked. If all those conditions exists then the element of the case is considered to be present (as it is deducted from the facts given by the user) and the system moves on to examine the remaining elements.

The user's answers are stored by the system in such a way that, when rules are used allowing, as conditions, facts that are common to rules which have already been, the system only requests facts for which a reply has not as yet been given.

Whenever the querying of the system gives an affirmative reply to the problem placed before it, LABEO asks the user if he is interested in checking whether the legal effects can result from using different rules. This could be particularly useful, examining which route is the most economical for reaching a particular legal result.

Whenever it is impossible, instead, to give an affirmative reply to the query (where the conditions required by some of the rules in the system which give that legal consequence are not present) and there is a rule establishing that the legal situation which was subject of the previous query does not exist as a legal consequence, the system asks the user if he is interested in knowing whether it is possible to exclude the fact that that effect has occurred.

If there is an affirmative reply, a new consultation is automatically initiated in which the subject of the query is the inexistence of the legal situation with which it deals.

Module No. 3:   Thesaurus Construction and Management.

Where there are numerous rules, the way to make the engine function more efficiently and therefore to facilitate the search for the solution is to minimize the space of the search. This requires a classification of the rules which reflect the relationship between the various legal effects.

For this purpose, the Thes-Maker program has been implemented. It allows whoever is responsible for the application to be assisted by the computer in all the stages of drafting, printing and consulting the special micro-thesauri of descriptors which should, for each different knowledge base, be constructed to facilitate the searching of the sentences and the phases of consultation of LABEO. The functions for the automated expansion of the inserted relations and for the permutation of the words contained in multiterm descriptors have proved particularly useful.

In the thesaurus for the LABEO the terms will be linked on the

basis of five relations: preference, hierarchy and association (expressed respectively by the USE, BT and RT operators) and the reciprocal relations of the first two of these (expressed by the UF and NT operators; there is no need to define a reciprocal operator for RT as T1 RT T2 is equivalent to T2 RT T1).

The "NOTE" relation which links a term to a note and is designed to clarify its meaning and use was introduced next.

Each relation has been assigned properties identified by reference to the semantic relations they aim at representing but they have also been defined in algebraic formulae so that they can be utilized by the computer for expanding and controlling these relations. The relations represented in the thesaurus are, therefore, the following:

T1 USE T2

The user, instead of using T1 which is not a term of the thesaurus, is advised to use its synonym, T2. The USE relation is irreflexive (there is no term T which would result in T USE T), asymmetrical (T1 USE T2 does not imply (T2 USE T1) and intransitive (T1 USE T2 and T2 USE T3 implies N(T1 USE 13).

T2 UF T1

The term T2 is a term of the thesaurus and the user is advised to use it instead of its synonym T1 which is a term in the thesaurus. The UF relation is reciprocal of the USE relation (T2 UF T1 <-> T1 USE T2) and is automatically expanded from the latter term (for each T1, T2 pair defined by the documentalist as T1 USE T2, the UF relation is automatically expanded to T2, T1 so that it becomes T2 UF T1).

T1 BT T2

The term T1 has as its immediate hierarchic superior the term T2. The properties of the BT relation are irreflexivity, asymmetry and intransitivity (the hyponymous relation which the BT relation aims at expressing is, instead, transitive, but we consider it preferable to limit the BT relation to hierarchically "contiguous" terms T1 and T2 or, in other words, that T2 is the directly superior term to T1).

T2 NT T1

The term T2 has as its hierarchic inferior the term T1. The NT relation is expanded automatically from its reciprocal BT (as T2 NT T1 <-> T1 BT T2).

T1 RT T2

The term T1 is associated with the term T2. The RT relation is irreflexive but is also symmetric and transitive. These latter properties are applied automatically whereby the RT relation is

expanded to those pairs of terms T1, T2 so that T1 RT T2 is logically derived from the RT relation defined by the documentalist (that is, the RT relation is expanded to those pairs of terms T1, T2 so that the properties of the RT relation are respected only where there is T1 RT T2). Therefore, the following situation exists:

based on symmetry:

T1 RT T2 -> T2 RT T1

based on transitivity:

T1 RT T2 and T2 RT T3 -> T1 RT T3.

T NOTE: n

The note n clarifies the meaning and the use of the term T. It is an irreflexive, asymmetric and intransitive relation.

For specifying the relational structure of the thesaurus, utilizing THES-MAKER, only the expansion of the USE, BT, RT and NOTE relations has been specified (we call expansion of a binary relation R the set of the pairs T1, T2 resulting from it so that T1 R T2), leaving the computer to expand the UF and NT relations and to develop the RT relation.

It should be noted that upto six terms T2 can be given for every T1 as T1 BT T2.

This means that each term can belong to different hierarchies (or trees) of concepts. The aim is thereby to avoid that kind of rigidity of which hierarchical thesauri are often accused.

It is also possible to specify upto six associated terms for each term.

Terms correlated with a given term T are recognized by their number (each term - descriptor or non descriptor - is given a number which unequivocally identifies it).

Specification of the relation is facilitated by consulting the computer drafted work lists. These lists contain, respectively, the alphabetic and numeric list of the terms and, for each individual term, its identification number and its relations with the descriptors in the thesaurus.

When T1 has the order number of another term T2 in the USE relation box (that is, a term T2 is such that T1 USE T2) the indexing of the more general or associated descriptors is prohibited (as T1 USE T2 means that T1 has not been accepted as a term in the thesaurus and that, for the semantic relations relating to T1, reference must be made to the synonymous descriptor T2).

357

After the relations between descriptors have been specified, the procedure for their checking and automated expansion can be started.

The vertical hierarchic relations between the terms (identified by the BT (Broader Term) operator and its reciprocal NT (Narrower Term)) formulated as algebraic tree structures, will have their formal properties expanded and checked automatically. A tree-structure with only one root and numerous intermediate and terminal nodes will be constructed for every set of hierarchic chains having the same conceptual origin. Then, by using the computer, each tree-structure will be checked to make sure there are no "jumps" in the hierarchy during the passage from one node to another (in the BT/NT chains) and that each node has only one hierarchic superior.

The horizontal relations of synonymy, quasi-synonymy and association between the terms (identified by the RT (Related Term) operator and expanded automatically according to their properties of symmetry and direct transitivity) will, instead, be designed as non-cyclic algebraic structures. They will be checked automatically in order to avoid any logical errors or redundancies. The computer will also verify whether the rule of incompatibility between BT and RT relations for the same pair of terms has been obeyed.

The fundamental methodological choice of giving each descriptor an identification number will make the various checking, processing and expansion procedures independent of the linguistic form used with the advantages of saving on storage space, of running the programs more rapidly and of facilitating the translation of the thesaurus into other languages.

Module No. 4: Knowledge Base Management.

The last module manages a series of service and control functions by those who are responsible for the system, including a) the choice of the knowledge base from among those which are available (up to a maximum of 26 bases); b) the up-dating of the knowledge base selected (which is increased automatically by the results of the normalization procedure described in Module No. 1) relative to the textual references, the sentences, the formulae, the descriptors and the texts in both their original and normalized versions; c) the production of reports and lists of the sentences and the formulas fed into the knowledge base in accordance with the options in ordering the data chosen by the user through the menu. Naturally, the user is able to print out either the entire knowledge base or only a parto of it, by inserting the conditions for selecting the part required.

* * *

LABEO: A KNOWLEDGE-BASED EXPERT SYSTEM

The first two modules have been developed in Turbo-Prolog which is a compiled programming language for artificial intelligence applications suitable for personal computers; modules 3 and 4 have been written in Clipper, a compiled language for data bases management. The supervisory environment in which the four modules described above interface has also been developed in Clipper.

3. CONCLUDING CONSIDERATIONS.

The LABEO experiment demonstrates - apart from the satisfying result reached in the normalized drafting of legal texts and in computer-assisted thesaurus building - that although we are still a fairly long way from the goal of constructing an evolved shell of an expert system based on consolidated and efficient techniques for the representation and processing of legal knowledge, we are a very long way from the concrete application, on a large scale, of such a system - even if it is theoretically possible - because it is not enough for its application that technical solutions have been found. Its implementation must also be economically viable and respond to an effective need in society and in the organization of traditional legal activities (those of the legislator and of other theoretical and practical legal operators).

And in relation to this last point there is likely to be a great deal of discussion before deciding what is desirable or what is compatible with the needs of the society and is in conformity with its norms and its ideals even if recently there has been an increased interest in the theory and practice of legislative techniques which should favour the advancement of similar research projects.

However, even if practical applications are never made (which occurred, instead, on a large scale for documentation retrieval systems), the theoretical interest in research of this kind will always be invaluable: the techniques utilized within the LABEO project may, in fact, not only suggest specific approaches to legal interpretation but may also help us in some way to understand the largely unconscious mechanisms behind legal interpretation, proposing models which can then be tested to the great benefit of the general theory of law and for teaching in our university law faculties.

Costantino CIAMPI

FIGURE 1
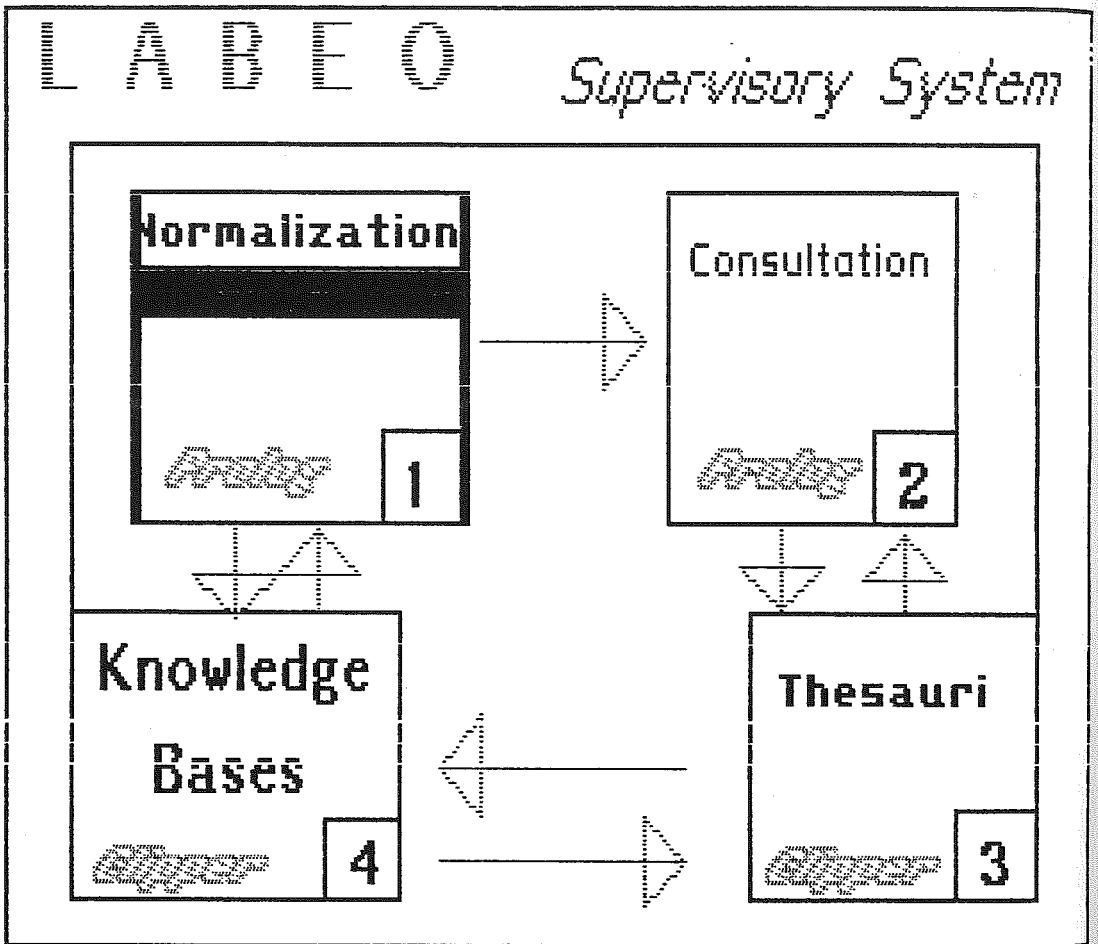
```
--------------------------------------------------------------
                        APPENDIX
--------------------------------------------------------------
```

Original Text:
(from Italian Building Indemnity Law No. 47 of 28 February 1985)

Art. 32 (1)
(Buildings Constructed in Restricted Areas)

    With the exception of the cases provided for in art. 33, the granting of a permit or authorization indemnifying buildings constructed in restricted areas, including therein those falling within national and regional parks, is subject to the approval of the administrations responsible for enforcing the restrictions. Whenever this approval is not given by the abovesaid administrations within one hundred and twenty days after the application has been made, it is to be understood that the administration's opinion has been negative.

Normalized Text:

1.   IF
     A. [ca]: the building is contrary to the restriction], AND
     B. [cc]: the restriction was imposed before its construction
     THEN
     C.   IF
          1.   IT IS NOT SO THAT
               [dfb4: the provisions of art. 33 of the Italian
               Building Indemnity Law are applied to the
               building],
          THEN
          2.   [dfb1: the indemnification of the building to
               which the restriction applies is subject to the
               approval of the administration responsible for
               enforcing the restriction] AND
2.   IF
     A.   [dfb1: the indemnification of the building to which the
     restriction applies is subject to the approval of the
     administration responsible for enforcing the restriction],
     THEN
     B.   IF
          1.   IT IS NOT SO THAT
               [bb1: the administration responsible for the
               restriction has given its approval]
          THEN
          2.   IT IS NOT SO THAT
               [dfb2: the building is capable of being
               indemnified] AND
     C.   IF

                                                           361
```

Costantino CIAMPI

1. IT IS NOT SO THAT
   [bb2: the approval was not given within 120 days],
THEN
2. [bb3: the approval is to be understood as negative]

## Interpretative Logic Formula:

$$(((ca1\&cc1)>(Ndfb4>dfb1))\&(dfb1>((Nbb1>Ndfb2)\&(Nbb2>bb3))))$$

## Arrow Diagram of the
## Logical Structure of the Text:

```
>—ca1—cc1 ——> Ndfb4 ——> dfb1
>dfb1——>——> Nbb1 ——> Ndfb2
       L—> Nbb2 ——> bb3
```

Research Director

Istituto per la Documentazione Giuridica

of the Italian National Research Council

Florence