A Knowledge-Based Systems Approach to Educating Creative Legal Minds

Hajime YOSHINO
Meiji Gakuin University Graduate Law School
1-2-37 Shirokanedai, Minato-ku
108-8636 Tokyo, Japan
+81 (3) 5421-6510
yoshino@ls.meijigakuin.ac.jp

Seichiro SAKURAI
Meiji Gakuin University Graduate Law School
1-2-37 Shirokanedai, Minato-ku
108-8636 Tokyo, Japan
+81 (3) 5421-6597
sakurai@ls.meijigakuin.ac.jp

ABSTRACT
Rapidly changing global societies of this century require us to have many good lawyers. "Good" lawyers must have creative legal minds. It is important to educate law students so as to enhance creativity as well as to ensure their knowledge of the law. We have, therefore, established a project on "the development of a legal education method for fostering creative legal minds." In this paper, we would like to present the results of one of our studies as an application of the legal knowledge based system developed by the authors to promote students creative legal minds. The basic idea is as follows: The traditional legal education in Japan, which belongs to the continental, statutory law culture, has been based on the lecture method, which uses a top-down implantation of legal knowledge. Our new approach involves an integration of four methods: problem-solving, Socratic interpretation and discussion, which together lead to a more interactive, bottom-up development of legal knowledge using the legal knowledge based system (LKBS). In this paper, the authors introduce a LKBS (LES4) they developed, discuss the applicability of the LKBS to legal education for creative legal minds and report an example of training of creative ability of students in the systematization of legal knowledge.

Categories and Subject Descriptors
L2.1 [Applications and Expert Systems]: Law and Legal Education.

General Terms
Design, Experimentation.

Keywords
Legal Expert System, Knowledge Based System, Creative Legal Minds, Law Education.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference Name, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/01... $5.00.

1. INTRODUCTION
Society is changing rapidly, and globalization is surging forward. It has become very important for legal practitioners to cope with such change in this time of intense transformation; that is, to provide appropriate solutions to legal problems that deal with the rapid progress of society.1 There are many criteria for being an excellent lawyer, but the capacity for creative and flexible legal thinking that is adaptable to this rapidly-changing society is especially in demand. With the backdrop of this era, cultivation of this creative legal thinking ability is offered as one of the vital educational objectives of law schools. We refer to education that cultivates this creative legal thinking ability as "creative legal education."

We have thought that the research (and results) of artificial intelligence could be applied toward this objective of implementing creative legal education. For this reason, we have brought to fruition some aspects of the development of the legal knowledge-base.2 We have thought, "Couldn’t we contribute to the realization of this objective utilizing these accomplishments, or developing and expanding them further?" The traditional legal education in continental law countries (sometimes referred to as civil law countries), to which Japan belongs, is based on lecture methods. In lectures, professors pass on to students the legal knowledge that they themselves possess, and the students have to understand and remember it. This is a top-down implementation of legal knowledge.

A new approach is an integration of problem methods, Socratic methods and discussion methods. Ultimately, it is designed to be a bottom-up development of legal knowledge, using a knowledge-based system.

1This is especially the case for Japan. There have been numerous cases in which Japanese companies lost legal disputes in the U.S. and other countries, causing Japanese legal professionals to become viewed as unreliable in comparison to their American counterparts. The demands of the financial world have pushed the political establishment to a systematic reform that will establish a graduate law school attempting to train a greater number of skilled lawyers.

2Legal Expert System LES-4th and others have been developed through this research project[1][2][3][4].
The establishment of legal propositions carried out through this sort of legal application process is referred to here as legal creation. Creative legal thought, then, is legal thought that establishes these kinds of legal propositions and resolves disputes appropriately.

2. APPLICABILITY OF LEGAL KNOWLEDGE-BASE TO CREATIVE LEGAL THOUGHT

Let us consider how the legal knowledge-base can be applied to creative legal education. The purposes of this application are divided into two main categories. One is the acquisition of fundamental knowledge and skills pertaining to legal thought as a premise for creative legal thought. The second is the promotion of the creative side of creative legal thought. It can be said that the former is the acquisition of the fundamental technique and basic form of legal thought necessary for performing the latter creative legal thought.

First, in order to properly acquire the fundamental knowledge and skill of legal thought, it is necessary on the one hand to understand the systematic structure of legal knowledge, and on the other hand, to practice actual legal thought through problem-solving case simulations. In the former, this is accomplished by (1) understanding the general logic system of legal knowledge, (2) understanding the general structure of legal reasoning, and (3) understanding the concrete knowledge structure of legal reasoning. In specific fields such as the Civil Code, the Commercial Code, and...

In order to establish legal knowledge, based on the understanding of the general logical system that accompanies the legal knowledge, it is necessary to carry out this creation by synthesizing two sides: the present side that embodies the meaning of the law, in order to apply it to concrete examples; and the side that systematizes legal knowledge by placing the various legal regulations in their systematic relationship, under the framework of the general legal knowledge logic system, in order to make the solutions of the various concrete examples justifiable. Better understanding is promoted through comprehension of both the general logical system and legal structure of legal knowledge and the basic framework of legal creation, and also through the application of legal knowledge to concrete problem-solving. It can be beneficial to use the knowledge-base system in this area.

3.2 Learning of systematization of legal knowledge

We previously explained that the creative side of legal interpretation is the methods of creating the meaning of the law and the methods of systematizing laws, but this paper will elucidate the creation side and focus on the systematization of legal knowledge.

Systematic legal thinking, is not something scattered among individual bits of legal knowledge, but rather, is something within its systematic relations. Legal specialists already possess, in addition to common-sense legal terms, the principle knowledge necessary to integrate legal knowledge, and in utilizing this having the ability to systematize and comprehend all kinds of legal knowledge's, especially legal articles.

Here, systematization is the construction of a deductive-reasoning system. This deductive-reasoning system is not a pre-existing entity determined beforehand through legal problem-solving, but is appropriated through a fixed framework and constructed in concrete detail in order to justify the problems' solutions. Furthermore, although the legal knowledge system is a dynamic system in contrast to the deductive-reasoning system of domains such as mathematics and natural science, it could be considered in certain situations, and from certain points of view, to have a deductive-reasoning system similar to that of natural science.

This systemic ability is the ability to construct a deductive-reasoning system. Deductive reasoning in law is not a given system. To systematize the law is not to recognize a given system, but to create a system. People who attempt to understand the law have to create it themselves. Actual legal reasoning in themselves do not constitute a strict system. People who apply and interpret the law systematize it themselves. The systematization of law is creative legal thought.

Students who aspire to be the legal professions must develop this systematic thinking ability in themselves. Law schools must also provide this training. In order to do this systematic legal thinking in students, first, it is useful to (1) explicitly present this general system structure of legal knowledge. Moreover, it is also effective to (2) actually put this legal knowledge to use: that is, to practice reasoning through the application of legal knowledge to problem-solving. In addition, (3) it is vital for the students themselves to practice systematization of legal knowledge. By carrying out the systematization themselves, students can develop systematic legal thought.

Utilization of the legal knowledge-base is effective for (1) and (2), while creation of the legal knowledge-base is effective for (3). We will explain (3) in more detail, namely by discussing examples of legal systematization by students applying the CISG to the hypothetical problems.

4. EXAMPLES OF TRAINING OF ABILITY IN LEGAL SYSTEMATIZATION

At first, the target domain is the CISG. Let us consider three types of examples, such as those in Table 1 below.

Here, certain articles of the CISG apply. In Figure 2 above, it is the systematization of legal creative reasoning that discovers the applicable portion of the legal principle.

The relevant CISG articles are as follows:

<table>
<thead>
<tr>
<th>Article 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) An offer becomes effective when it reaches the offeree.</td>
</tr>
<tr>
<td>(2) An offer, ..., may be withdrawn if the withdrawal reaches the offeree before or at the same time as the offer.</td>
</tr>
</tbody>
</table>

Inclusively, this is taught using six examples, but for ease of explanation, we will simplify the demonstration by only using three examples.

---

This knowledge includes meta knowledge related to the application of knowledge. It can be thought that systematic legal thought may be the ability to compose reasoning of deductive justification using this meta knowledge.
5. A SOFTWARE SYSTEM FOR SUPPORTING OUR EDUCATION METHOD

This section describes a software system for supporting our education method. The software system is currently being implemented on a Web application server.

5.1 System overview

While our education method needs an ability to construct a knowledge-base, it is not so easy task for a student as a novice programmer. To alleviate the task, an editor and an inference engine have been designed on a Web application server. As a Web application server, Zeop[6], which is an open source Web application server written in python, is selected as a pilot home. Zeop is composed by ZODIR, which is an object oriented DIR manager and Zopublisher, which outputs HTML files from ZODIR. Plenty of knowledge-bases, which are generated or abandoned by students in our education method, can be stored in ZODIR as objects. Furthermore, Plex[7], which is a product to construct a Portal site on Zeop, is used to manage user accounts. By using Zope/Plex, the learning history of each user will be accumulatated. Since the accumulated learning history can be used for analysis of learning effect, example problems or instructions strategies will be refined based on the analysis.

5.2 Structured editor on Web

For a student of graduate law schools, it is not so easy to understand the syntax of a logic program such as Prolog and the notion of "term." If all of students know the syntax and notion of "term," our method could be directly applicable. However, since few of students are assumed to have programming knowledge, some assistant systems should be prepared. In such a system, editing knowledge should be utilized.

Since HTTP is mainly implemented by only POST and GET operations, the interaction between the server and the user is very restrictive to implement an editor. A structured editor is well-known as an editor for Lisp programs and it works on telnet terminals. In other words, the structured editor of Lisp works well on a half duplex communication line. The behavior of the structured editor is very simple. It only reads the user's command and writes the result of user's command. Since the requirement portion of a rule proposition can be represented as a list data structure10, the structured editor of Lisp is suffice to edit the requirement portion of a rule proposition. The current commands are (1) deleting an expression, (2) introducing a conjunction and (3) introducing a disjunction, and (4) introducing a negation. After editing rule and fact propositions, students must designate a collection of rule and fact propositions. Since Zeop has an object oriented ZODIR, a hierarchical knowledge-base can be easily constructed by using the inheritance function.

5.3 Inference Engine

After the temporary construction of a knowledge-base, the correctness of the knowledge-base can be verified through actual

---

10 In the current prototype implementation, only function free logic programs are considered.
simulations. In order to solve or trace the given query, an inference engine is needed. As all of interfaces are on a Web application server, light weight inference engine is desirable. PyLog[8], which is a Prolog interpreter written in python, converts a Prolog program into python programs. As the knowledge-base is compiled into a python program, the query with respect to the knowledge-base can be directly executed as a server side script like ASP, JSP or PHP. If the query is solvable successfully, its proof tree will be presented to the user like LESH-5[1][2][3][4][5]. As a result, students revise their knowledge-bases according to the result of the inference engine.

6. CONCLUSION

Legal study up to this point has been mostly a matter of the professor simply explaining law by presenting students with legal knowledge acquired through many years of study or research and telling them, “Here’s what the law is.” The students would imbibe that teaching, undergo rote learning, and for some students it would just go in one ear and go out the other. At any rate, students’ attitudes toward education were continually passive, and their ability to creatively think for themselves was poor. Thus, the traditional legal teaching methods, which use a top-down approach, have been mostly unsuccessful.

In educational training of legal systematic thought, utilizing the legal knowledge-base system, as outlined in this paper, through application to multiple examples and problem-solving exercises, and by constructing the legal knowledge-base, students can learn and develop creative systematic thinking. Consequently, this legal knowledge-base system is designed to operate from the bottom-up, in contrast to the traditional top-down teaching methodology.

We began partly testing this educational approach in both the “Legal Method” class and “Law and Artificial Intelligence” class at Meiji Gakuin University graduate law school, which opened in April 2004; we also used this approach in the “Law and Computers” class at Keio University School of Law. We would like to say that upon its actual utilization in a classroom setting, law school students began to find their own mistakes and gradually corrected them to achieve better systematicization rules. Students’ reaction have been positive, including refreshed interests, and, overall, students have found this training useful in learning and acquiring systematic knowledge and general structure. Consequently, we believe that its effectiveness has, to some degree, been confirmed in the educational domain.

Regarding systematic thinking, aside from the systematization issues of legal articles and concepts explained in this paper, there is also a need for legal thought that inductively discovers laws (principles) from examples of individual judgments. Training in this legal thinking ability is the goal of education that uses case method and the Socratic Method, especially in countries that practice common law (England, the United States, etc.).

Introduction of this training is currently being tested in our country, especially in law schools. The application of information technology, especially artificial intelligence theory and technology, to this education is in demand. We are pushing forward with this experiment. Systems supporting the Socratic Method were touched on in another report. We would like to introduce education of legal deduction reasoning in a future manuscript.

The aforementioned matter is the systematization side of legal interpretation; of course, legal creation education is not on this systematization side, but rather the embodiment side, and in this area, having students create embodiment rules for concrete example problem-solving is not only possible, but also essential. The legal education base, or the construction of the legal education base, is also thought to be effective toward this purpose. Educational utilization on this side will become a key issue in the not-too-distant future.

7. REFERENCES


An Integration of a Legal Knowledge Based System and a Content Management System for a Legal Education Support System

Seiichiro SAKURAI           Hajime YOSHINO
Meiji Gakuin University Graduate Law Schol
1-2-37 Sirokanedai, Minato
108-8636 Tokyo, Japan
{sakurai,yoshino}@ls.meijigakuin.ac.jp

Abstract
This paper describes an integration of a knowledge based system and a content management system (CMS) for a legal education support system. The authors' education method utilizes a legal knowledge based system to help students to acquire the structure of law from statues in a bottom up manner. From the viewpoint of the acquisition of creative legal minds, erroneous knowledge bases are also needed to understand why the erroneous knowledge bases are erroneous. To provide only correct knowledge results in rote learning. For creative lawyers, thinking by themselves is the most important task. Therefore, in our approach, students must select or construct their correct knowledge bases from many alternatives through the comparison. In order to support our approach, a legal education support system should handle many separated knowledge bases which are stored in a database. This paper proposes a method for implementing a number of small knowledge bases on a CMS.

1. Introduction
To encourage the ability of creative thinking is one of the major roles of Japanese law schools. In order to avoid the rote learning of law, the authors have proposed an education method[5] which uses a knowledge based system. By comparing the inferred results of knowledge bases, students can find why a knowledge base is superior to others. However, since the traditional implementation of a knowledge based system is so heavy, simultaneous use of the knowledge based system may cause the degradation of the system performance. If all members of a class access the knowledge based system, the knowledge based system may stop. To solve the degradation of the system performance, this paper proposes a system design by using a light weight Prolog implementation.

A content management system (CMS) is very important technology for a Web based application server because of the
separation of contents and their representations. If a Web based
application server does not have a CMS,
the maintenance of contents of the server
becomes very hard. Therefore most of e-learning systems have a function of a
CMS or are constructed on a CMS. In
order to promote creative thinking,
embedding a rule based inference system
into a CMS is also an important issue. In
our education method, a number of small
knowledge bases are utilized to educate
legal knowledge through the interaction
with law students.

Consequently, the efficiency and the
scalability are the main issues in this
paper. These issues can be solved by a
light weight Prolog implementation and
small objects of knowledge bases.

2. Content Management System
Zope[3] is a Web application server
written in Python[1], which is a modern
script language, and it functions as a Web
server. In Figure 1, the Web application
server is realized by Zope. Since all of
contents of Zope are stored in a database,
Zope dynamically generates HTML files
according to users requests.

![Figure 1: a Web Application Server with a CMS](image)

While Zope outputs HTML files basically,
Plone[2], which is a CMS product for Zope,
can manage many kinds of contents such
as documents, multimedia files, md so on.
Furthermore, Plone has many functions
for a Portal Web site. From the viewpoint
of a system programmer, one of the most
important features of Plone is its
extensibility. Since Plone is easy to
extend, many additional products such as
BBS, assignment database or Weblog can be
installed easily on Plone. Actually, the
authors are implementing a Socratic
methods support system on Plone.

3. Inference Engine Written in Python
The basic idea of authors’ education
method is to let students think the
meaning of legal provisions by
themselves. Since most of legal provisions
are related to other legal provisions or
legal concepts, in order to understand the
meaning of legal provisions, the students
must find hidden relations of legal
provisions. The process of finding hidden
relations is basically by trial and error.
Therefore, if the meaning of legal
provisions is assumed to be represented
by rules, many rules, including erroneous
rules, are generated and abandoned. In
order to ease the burden, providing many
rule sets as knowledge bases can be
considered as a solution. In that case,
erroneous knowledge bases are useful to
understand why some knowledge base is
superior to the erroneous one.

One of the merits of the knowledge based
system is that the knowledge based
system can be used to verify the inferred
results of a knowledge base. It is not so
useful for students to view a knowledge
base because of their ignorance of the
relations of rules. In order to help
students to understand hidden relations
of rules, it may be useful to present the
inference result and inference trace of
knowledge bases as shown in Figure 2.
By comparing the results or inference traces, students can understand why
some knowledge base is superior to
others.

![Figure 2: Multiple Knowledge Bases](image)

An inference engine is the most
important part of a knowledge based
system. The basic inference engine of rule
based system is well known as
SLD-resolution[6] and its well-known
implementation is Prolog[7].

PyLog[4] is a Prolog interpreter written
in Python[1]. In PyLog, since the depth
first search of Prolog[7] is elegantly
implemented by using the “generator”
feature of Python, Prolog codes can be
run efficiently
The “generator” feature of Python is a control structure like a
thread, which is well known as a device of
a light weight process, and the generator runs much faster than the usual thread.
Thus Prolog programs can be run only
with little overhead of term unifications
\footnotemark[5]

\footnotetext[5]{While one of the most important
features of Prolog is the backtracking
mechanism, if the completeness of the
solution search is guaranteed, the
backtracking mechanism is not always
needed for an inference engine.

\footnotetext[6]{The unification algorithm without
occurrence checking is known to be very
}
As the Prolog codes are automatically translated into Python classes, any knowledge written in rules can be embedded in a Python program. The most important feature of PyLog, which is at most 2000 lines, is its compactness. Thus the invoking overhead is very small for Plone/Zope. This fact means that our implementation is also effective to other CMS with the ability to invoke external programs. For example, since Python is also implemented on Java, our method is effective to Java-based CMS.

4. Examples of Knowledge Bases

Each rule is stored in the database as an object, which has two elements, requirement and effect. Both of the requirement and the effect are character strings which denote predicate names, logical variables and logical connectives. For example, a rule, "if p(x) and q(x) then r(x)" are coded in as follows:

Effect: "r(X)".
Requirement: "p (X) AND q (X)".

The rule object such as above is the primitive of a knowledge base. A knowledge base can be defined as an object which has pointers to actual rule objects. By defining the knowledge base as a small object, the memory space of the knowledge bases is in proportion to the number of knowledge base.

When a user requests the result or the inference trace of a knowledge base, the knowledge base is automatically compiled into Python classes by using PyLog. For example, the above rule is compiled into the code in Figure 3.

class r(PyLogPredicate):
def __init__(self, a0):
    self_form_args = Term(a0)
    X = Var('X')
    self_exprs =
    [(Term(X), lambda:And(q(X),r(X)))]

Figure 3: Python Class of Prolog Code

Since Prolog predicate "r", which is shown in Figure 3, is defined as a subclass of PyLogPredicate, the provability of "r(X)" can be confirmed by invoking "r(X)" in any Python scripts.

In this way, the workable knowledge base can be stored in the working memory as Python executable codes. This fact can also guard the knowledge base against the leaks.

5. Related Works

Several attempts of implementing Prolog by using client side programming have been done. Since one of the most popular client side programming languages is JavaScript or ECMAScript, some implementations of Prolog are available on the Internet. While the Prolog written in ECMAScript[10] is very portable in the sense that most of Web browsers can execute ECMAScript codes, it may leak the program codes of knowledge bases because ECMAScript codes are parts of a HTML file. In contrast to the Prolog written in ECMAScript, each knowledge base can be guarded by the security policy of Plone in our method. The mode of a knowledge base, which indicates whether the knowledge base is executable or viewable, can be easily controllable by the authors of the knowledge base. This is one of the merits of the combination with a CMS.

Another client side implementation can be considered as Java applets[8,9]. Since the Java applets are restrictive with respect to the communication with the server because of the security, a naive implementation can be than a Java applet must have only a knowledge base. If the number of knowledge bases is small, the naive approach seems to be a candidate implementation. However, if the number of knowledge bases is big, the cost of downloading the Java applets may be expensive. Therefore, especially in case of ours, the naive Java applet approach is inappropriate because of the number of knowledge bases.

6. Conclusion

This paper clarifies the implementation issues of our education method. In our method, since each knowledge base must be confirmed by students that it is superior to any other knowledge bases, the support system must store many separated knowledge bases and infer the result of a selected knowledge base. In order to solve the issues, a knowledge based system is embedded in a CMS. Currently, the legal education support system is being constructed.

References