ICOT Technical Memorandum: TM-1299

TM-1299

A Legal Reasoning System based on Situation theory

by

S. Tojo (MRI), K. Nitta & S. Wong (UCSF)

June, 1994

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5

Institute for New Generation Computer Technology

A Legal Reasoning System based on Situation Theory

Satoshi Tojo
Mitsubishi Research Inst., Inc.
tojo@mri.co.jp

Stephen Wong
UCSF
swong@lri.library.ucsf.edu

Abstract

Legal reasoning systems research is a new field attracting both AI researchers and legal practitioners. The purpose of this paper is to introduce a formal model of legal reasoning, based on situation theory. On that abstract model, we show an example of reasoning system implemented in a knowledge-base management language *QUIXOTE*, regarding the language as a situated inference system.

1 Introduction

Legal reasoning systems research is a new field which has attracted researchers ¿from both the legal and AI domains. Most legal reasoning systems draw arguments by interpreting judicial precedents (old cases) or statutes (legal rules), while more sophisticated systems include both kinds of knowledge. Surveys of the leading projects can be found in [4, 5].

Thus far, those legal reasoning systems seem to have had weak foundation in formalization, and they have been *ad hoc* combination of various forms of logical inference. Our prime purpose of this paper is to give a sound foundation to legal reasoning system in terms of situation theory [1]. And, in addition, we implement this model into a computational form in Knowledge Base Management System (KBMS) *QUIXOTE* [7]. Regarding the concept of *module* of *QUIXOTE* as situation, we show that the language can work as a situated inference system. The set of knowledge bases includes a dictionary of legal ontologies, a database of old cases, and a database of statutes.

The organization of this paper is as follows. Section 2 describes the formulation of legal knowledge at the abstraction level using the theory of situations. Section 3 illustrates the realization of this formulation at the KBMS level using *QUIXOTE*, and its situated inference mechanisms. The last section concludes this paper.

2 Situation Theory for Legal Reasoning

This section introduces a formal model for legal reasoning, especially, penal code, at the abstraction level. The formulation is based on *situation theory*, so we call it a situation-theoretic model ($\mathcal{SM}$).

2.1 General Terms

The ontologies of $\mathcal{SM}$ include objects, parameters, relations, infons, and situations. An object designates an individuated part of the real world: a constant or an individual in the sense of classical logic. A parameter refers to an arbitrary object of a given type. An n-placed relation is a property of an n-tuple of argument roles, $r_1, \cdots, r_n$, or slots into which appropriate objects of a certain type can be anchored or substituted. For example, we can define 'eat' as a four-place relation of *Action* type as:

$< \text{eat:}Action \mid eater{:}ANIMAL,\ thing\text{-}eaten{:}EDIBLE\text{-}THING,\ location{:}LOC,\ time{:}TIM >$

where *eater, thing-eaten, location,* and *time* are roles and the associated types, *ANIMAL* denotes the type of all animals, *EDIBLE-THING* denotes the type of all edible substances, and *LOC* and *TIM* are types of spatial and temporal location.

An infon σ is written as $\ll Rel, a_1, ..., a_n, i \gg$, where Rel is a relation, each argument term a_k is a constant object or a parameter, and i is a polarity indicating 1 or 0 (true or false). If an infon contains an n-place relation and m argument terms such that $m < n$, we say that the infon is *unsaturated*; if m = n, it is *saturated*. Any object assigned to fill an argument role of the relation of that infon must be of the appropriate type or must be a parameter that can only anchor to objects of that type. An infon that has no free parameters is called a *parameter-free* infon; otherwise, it is a *parametric* infon. If σ is an infon and f is an *anchor* for some or all of the parameters that occur freely in σ, we denote, by $\sigma[f]$, the infon that results by replacing each v in the domain of f that occurs freely in σ by its value (object constant) $f(v)$. If I is a set of parametric infons and f is an anchor for some or all of the parameters that occur freely in I, then $I[f] = \{\sigma[f] \mid \sigma \in I\}$.

$\mathcal{SM}$ is a triplet $\langle \mathcal{P}, \mathcal{C}, \models \rangle$, where $\mathcal{P}$ is a collection of abstract situations including judicial precedents, a new case, c_n, and a world, w, that is a unique maximal situation of which every other situation is a part. $\mathcal{C}$ is a concept lattice in which objects are introduced and combined with the *subsumption* relation ('=<'), that is an *is-a* relation intuitively, each other. 'An object of a type' is interpreted as 'an object is subsumed by another object corresponds to that type'. '$\models$' is the support relation, and our interpretation is:

Definition 2.1 (Supports Relation) *For any $s \in \mathcal{P}$, and any atomic infon σ, $s \models \sigma$ if and only if (iff) $\sigma \in s$.* □

2.2 Situated Inference Rules

Reasoning in law is a rule-based decision-making endeavor. A legal reasoning process can be modeled as an inference tree of four layers. The bottom layer consists of a set of basic facts and hypotheses, the second layer involves case rules of individual precedents, the third layer involves case rules which are induced from several precedents or which are generated from certain legal theories, and the top layer concerns legal rules from statutes. An individual or local case rule is used by an agent in an old case to derive plausible legal concepts and propositions. These rules vary from case to case, and their interpretation depends on the particular views and situational perspectives of the agents. An induced case rule has a broader scope and is generalized ¿from a set of precedents. Legal rules are general provisions and definitions of crimes. They are supposed to be universally valid in the country where they are imposed, and neutral. That is, the applicability of these rules is independent from the view of either side (plaintiff or defendant) and every item of information (infon) included is of equal relevance. Though it rarely happens, it may be possible for an agent to skip one or two case rule layers in attaining a legal goal.

In such a rule-oriented legal domain, *situated inference* has the following general form:

Rule 1 (General Rule) $s_0 \models \sigma_0 \Leftarrow s_1 \models \sigma_1, s_2 \models \sigma_2, ..., s_n \models \sigma_n/B$,
where $\sigma_0, \sigma_1, ..., \sigma_n$ *are infons, and* $s_o, s_1, ..., s_n$ *are situations.* □

This rule can be read as: "if s_1 supports σ_1, s_2 supports σ_2, and so on, then we can infer that s_0 supports σ_0 under the background conditions or constraints B." $s_0 \models \sigma_0$ is called the head of the rule while the remainder is called the body of the rule. The background conditions, B, are required to be coherent and satisfied before execution of the rule. Note that $c \models I/B$ implies that $c \cup B \models I$, where $c \models I$ as a shorthand for $c \models \sigma_1, c \models \sigma_2, ..., c \models \sigma_n$.

We are particularly interested in three rule instances: local case rules, induced case rules, and legal rules. A local rule is as follows:

Rule 2 (Local Rule) *For* $c \in \mathcal{P}$, $cr : c \models \sigma \Leftarrow c \models I/B_{cr}$. □

where I is called the antecedent of the rule, σ is the consequent, and cr is the label of the rule, which is not part of the rule but which serves to identify the rule. Sometimes, we simply write cr: $c \models \sigma \Leftarrow I/B_{cr}$. Both σ and I are parameter-free. One unique feature of rules in the legal domain is that the consequent is not disjunctive and often a single predicate. The reliability and the scope of application of a local rule will be subject to a set of *background conditions*, B_{cr}. The conditions include information such as an agent's goal and hypotheses; these are crucial in debate to establish the degree of certainty and the scope of applicability of that rule. Usually, it becomes necessary to take background conditions into account and investigate what they are. Many case rules exist in one case and often yield incompatible conclusions. But, the background conditions clarify their hypotheses and perspective. When there is no danger of conclusion, we can write such a rule without stating its background conditions.

Another form of case rule is generalized or induced from several precedents. Owing to its generic nature, an induced case rule is represented as a constraint between two parametric infons, rather than parameter-free ones. Denote I' and σ' as a set of parametric infons and a parametric infon, respectively, such that all parameters that occur in the latter also appear in the former. An induced rule is written as:

Rule 3 (Induced Rule) *For any* $c_1, ..., c_k \in \mathcal{P}$, $c = c_1 \cup c_2, \cup ... \cup c_k$, $ir: c \models \sigma' \Leftarrow I'/B_{ir}$. $\square$

where c is coherent and ir is the rule label. Similarly, a legal rule is:

Rule 4 (Legal Rule) $lr: w \models \sigma' \Leftarrow I'/B_{lr}$. $\square$

where lr is the rule label and B_{lr} states the background legal theory, such as the aim of punishment or the aim of crime prevention, but not both. Such information is crucial in interpreting the antecedent infons.

2.3 Substitution and Anchoring

When a situation of a new case, c_n, supports a similar antecedent of a local rule of c_o, one can draw a conclusion about the new case that is similar to the consequent of that rule.

Definition 2.2 (Local Rule Substitution) *For* $c_n, c_o \in \mathcal{P}$, $cr^s: c_n \models \sigma\theta$ *if* $cr: c_o \models \sigma \Leftarrow I/B_{cr}$ *and* $c_n \models I'/\{B_{cr}\theta \cup B_n\}$ *such that* $I' \simeq_s I$. $\square$

where cr^s is the label of the new rule, B_n is the original background of I' of the new case, and the combined condition after the substitution, $B = B_{cr}\theta \cup B_n$, is coherent. The notation $\simeq_s$ denotes the matching relation between two situations. Section 3 discusses how such a matching is implemented in $\mathcal{QUIXOTE}$. The function θ forms a *link* that connects c_n with c_o. This function replaces all terms (objects and relations) in σ and B_{cr} that also occur in I with their matched counterparts in I'. Normally, the background conditions are not included.

To combine the conclusions supported by different situations, the background conditions of both conclusions must be compatible. That is why the background conditions of Rule 1 must be coherent. Rather than substitution, a consequent is derived from a legal rule or an induced rule via anchoring.

Definition 2.3 (Induced Rule Anchoring) *For* $c_n, c_1, ..., c_k \in \mathcal{P}$ *such that* $c = c_1 \cup c_2 \cup ... \cup c_k$, $ir^a: c_n \models \sigma[f]$ *if* $ir: c \models \sigma \Leftarrow I/B_{ir}$ *and* $c_n \models I[f]/\{B_{ir}[f] \cup B_n\}$. $\square$

Definition 2.4 (Legal Rule Anchoring) *For* $c_n \in \mathcal{P}$, $lr^a: c_n \models \sigma[f]$ *if* $lr: w \models \sigma \Leftarrow I/B_{lr}$ *and* $c_n \models I[f]/\{B_{lr}[f] \cup B_n\}$. $\square$

2.4 Matching of Infons and Situations

In order to compare the similarity of a new case with precedent cases, we formalize the infon matching and the situation matching. Suppose that a concept lattice is given, where the subsumption relation ('=<') is defined between concepts. $R(\sigma)$ is a function that extracts '*rel*' from an infon σ.

Definition 2.5 (Infon Matching) *For any two infons σ_1 and σ_2,*

1. *If there is a $R(\sigma_3)$ such that $R(\sigma_1)$ =< $R(\sigma_3)$, and $R(\sigma_2)$ =< $R(\sigma_3)$ in a given concept lattice, then σ_1 and σ_2 are interpreted as* weakly matched infons.

2. *If $R(\sigma_1) = R(\sigma_2)$, then then σ_1 and σ_2 are interpreted as* partially matched.

3. *If all the objects that constitute two infons are identical, then the infons are* exactly matched. □

We give concepts of situation matching below. Note that the concept of situation matching is independent of that of infon matching.

Definition 2.6 (Situation Matching) *For situations s_1 and s_2,*

1. *If, for every infon in s_1, there is an infon that can match it in s_2, and vice versa, then the two situations are interpreted as* exactly matched *situations.*

2. *For any σ_1 in s_1, there is an infon σ_2 in s_2 that can match σ_1, situation s_1 can be* partially matched *with situation s_2. (Note that this partially matching relation is one way; even though s_1 can be partially matched with s_2, s_2 may not be partially matched with s_1.)*

3. *For any σ_1 in s_1 whose* relevance *value is larger than a given threshold level, there is an infon σ_2 in s_2, that can be matched with σ_1, s_1 can be* partially matched *with s_2* w.r.t. relevance value. □

Among several matching definitions, we will adopt *weakly matching* for infons and *partially matching w.r.t. relevance value* for situations, in implementation of the following section, for practical reasons. Let us consider the following pair of descriptions:

$$\begin{aligned} s_{new} &\models \{\ll abandon, mary^{agent} \gg, \\ &\quad \ll leave, mary^{agent}, june^{object} \gg\} \\ s_{old} &\models \{\ll abandon, jim^{agent}, tom^{object}, 3^{relevance} \gg, \\ &\quad \ll leave, jim^{agent}, tom^{object}, 2^{relevance} \gg, \\ &\quad \ll poor, jim^{agent}, 1^{relevance} \gg\} \end{aligned}$$

If the threshold value is 2, then s_{new} can be *partially matched* with s_{old} *w.r.t. the value 2.* On the other hand, if 1 is the threshold instead, s_{new} cannot be *partially matched* with s_{old} *w.r.t. the value 1.*

In fig. 1, the combination of facts, case rules, induced rules, and legal rules is depicted.

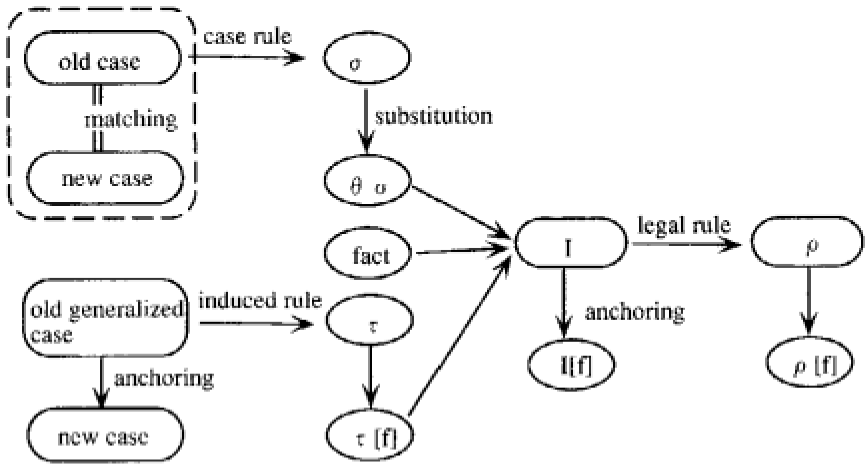

Figure 1: The combination of rules

3 Modeling of Legal Knowledge in $\mathcal{QUIXOTE}$

This section introduces the language of $\mathcal{QUIXOTE}$ [7] and shows how this language can represent the $\mathcal{SM}$ concepts in computable form. A typical $\mathcal{QUIXOTE}$ database includes the following data structures: (i) the subsumption relations among basic objects, (ii) the submodule relations among modules, and (iii) rules. Our legal reasoning system consists of three databases: a legal dictionary, cases, and statutes. Accordingly, we first introduce the *objects* and *modules* of $\mathcal{QUIXOTE}$ and explain the data structure of the legal dictionary, then describe the use of $\mathcal{QUIXOTE}$ *rules* to represent case-based rules and statutes.

In $\mathcal{QUIXOTE}$, the concepts of $\mathcal{SM}$ are rephrased as follows:

$\mathcal{SM}$	$\mathcal{QUIXOTE}$
situation	module
infon	attribute term
relation name	basic object
type	subsumption
role	label
supporting relation ($\models$)	membership in module (:)

3.1 Description of Case and Rule

A $\mathcal{QUIXOTE}$ rule has the following form (compare with Rule 1):

$$\overbrace{m_0 :: H}^{head} \mid \overbrace{HC}^{head\_constraints} \Leftarrow \overbrace{m_1 : B_1, \ldots, m_n : B_n}^{body} \parallel \overbrace{BC}^{body\_constraints} \quad ;;$$

where H or B_i are objects, and HC and BC are sets of formulas (constraints) using subsumption relations. Intuitively, this means that if every B_i holds in a module m_i under the constraints BC, then H and constraints HC hold in m_0. The head constraints and module identifiers can be omitted, and the body constraints, BC, of a rule then constitute the background conditions for that rule.

This study regards a case as being a situation, that is, a set of anchored sentences. Below, we describe a case which is a simplified description of an actual precedent [3].

Mary's Case

On a cold winter's day, Mary abandoned her son Tom on the street because she was very poor. Tom was just 4 months old. Jim found Tom crying on the street and started to drive Tom by car to the police station. However, Jim caused an accident on the way to the police station. Tom was injured. Jim thought that Tom had died in the accident and left Tom on the street. Tom froze to death.

This aforementioned case contains some human objects and several events with different relevancy. The order of values of the relevance attribute is represented by a subsumption relation, (`l1 =< l2 =< l3`).

```
mary_case :: {mary, tom, jim, accident, cold},
 poor/[agent=mary, relevance=l1],
 abandon/[agent=mary,
         coagent=tom/[mother=mary, age=4months], relevance=l2],
 find/[agent=jim, object=tom/[state=crying], relevance=l1],
 make/[agent=jim, object=accident, relevance=l2],
 injure/[agent=jim, coagent=tom, by=accident, relevance=l2],
 leave/[agent=jim, coagent=tom, relevance=l3],
 death/[agent=tom, cause=cold, relevance=l3]};;
```

The attorneys on both sides interpreted Mary's case according to individual perspectives: one is the responsibility of Mary's actions and the other is that of Jim's. For instance, one attorney reasoned that: "If Mary hadn't abandoned Tom, Tom wouldn't have died. In addition, the cause of Tom's death is not injury but freezing. Therefore there exists a causality between Tom's death and Mary's abandoning."

Another lawyer, however, argued differently: "A crime was committed by Jim, namely, his abandoning Tom. And in addition, Tom's death was indirectly caused by Jim's abandoning Tom. Therefore, there exists a causality between Tom's death and Jim's abandoning."

For a legal precedent, these contradictory claims are documented together with the final verdict from the judge overseeing that precedent. *QUIXOTE* models these arguments with two *case rules* of different interpretations of causality.

```
cr1 :: responsible/[agent=mary,for=death]
        <=
       abandon/[agent=mary,coagent=tom],
       death/[agent=tom, cause=abandon/[agent=mary,coagent=tom]];;

cr2 :: responsible/[agent=jim,for=death/[agent=tom]]
        <=
       leave/[agent=jim, coagent=tom],
       death/[agent=tom, cause=leave];;
```

The idea of an *induced rule* is to abstract some of ground terms in local case rules. As an example, when there are several similar accident cases, the attorneys may make the following generalization:

```
ir1 :: responsible/[agent=X, to=Y, for=Inj]
        <=
       Acc/[agent=X],
       Inj/[agent=Y, cause=Acc]
       || {Acc =< accident, Inj=<physical_damage,
       X =< person, Y =< person};;
```

In `ir1`, traffic accident and injury are abstracted to variable `Acc` and `Inj` and subsumed by their super concepts in the legal dictionary.

Legal rules, or statutes, are formal sentences of codes. We provide a penal code in linguistic form (Japanese penal code, article 199): "In case an intentional action of person A causes the death of person B and the action is not presumed to be legal, A is responsible for the crime of homicide."

The *QUIXOTE* representation of this code is:

```
lr1 :: responsible/[agent=A, to=B, for=homicide]
        <=
       Action/[agent=A],
       illegal/[act->Action],
       death/[agent=B, cause->Action],
       || {Action =< intend, A =< person, B =< person};;
```

In the description above, `illegal[agent=A, action = Action]` claims that the action `Action` done by `A`, such as self-defense, is not legal. The statute for the legality of self-defense is described as follows (Japanese penal code, article 38):

```
lr2 :: illegal/[act = Action]
        <=
       Action,
       || {Action =< intend};;
```

The concept of *anchoring* of $\mathcal{SM}$, mentioned in Section 2.3, is realized in $\mathcal{QUIXOTE}$ by invoking appropriate rules within a case or statute description.

3.2 Query Processing

Let us consider Mary's case, where $\mathcal{QUIXOTE}$ draws several conclusions by making different assumptions. In response to the query:

```
?-responsible/[agent=jim, to=tom, for=homicide].
```

that means "Is Jim responsible to Tom for the crime of homicide?", $\mathcal{QUIXOTE}$ returns the following:

```
** 2 answers exist **
** Answer 1 **
 IF mary_case:death.cause =< leave THEN
 YES
** Answer 2 **
 IF mary_case:death.cause =< traffic_accident THEN
 YES
```

The first answer is one interpretation of the causality in Mary's case: if the cause of Tom's death is some event under Jim's leaving Tom, then Jim is responsible for the homicide. The latter answer says that Jim is responsible if Tom had been killed by Jim's traffic accident. It happens, however, that the latter does not hold, so that the inquiring agent starts a new query which adds information about the cause of Tom's death.

```
?-mary_case:responsible || {mary_case:death.cause==leave}.
```

In response to this second query, the $\mathcal{QUIXOTE}$ system replies as follows.

```
** 1 answer exists **
** Answer 1 **
 IF mary_case:death.cause == leave THEN YES
```

Thus, we have shown the implementation of our situated inference model in $\mathcal{QUIXOTE}$.

4 Conclusion

In this paper, we formalized legal reasoning in terms of $\mathcal{SM}$, where precedent cases and new accidents were regarded as situations, and various kinds of rules as situated inference rules. We also showed that the abstract model was implemented in $\mathcal{QUIXOTE}$ for prototyping. $\mathcal{QUIXOTE}$ could represent context-dependent knowledge and situated inference for knowledge base applications. The ability of $\mathcal{QUIXOTE}$ to model abstract concepts of situation theory in a database environment may pave the way for the knowledge-base (KB) community to tackle concrete, demanding problems, such as building a large scale KB for general linguistic concepts.

References

[1] J. Barwise, *The situation in logic*, CSLI Lecture Notes 17, Stanford, CA, 1988.

[2] K. Devlin *Logic and information*, Cambridge University Press, 1991.

[3] K. Nitta, Y. Ohtake, S. Maeda, M. Ono, H. Ohsaki, and K. Sakane, "HELIC-II: A legal reasoning system on the parallel inference machine," *Proc. Int. Conf. of Fifth Generation Computer Systems*, ICOT, Tokyo, June, 1992, pp. 1115-1124.

[4] E.L. Rissland, (ed.), Special issue: AI and Legal Reasoning, Part 1, *International Journal of Man-Machine Studies*, Vol. 34 No. 6, June 1991.

[5] E.L. Rissland, (ed.), Special issue: AI and Legal Reasoning, Part 2, *International Journal of Man-Machine Studies*, Vol. 35 No. 1, July 1991,

[6] S. Wong, "A situation-theoretic model for trial reasoning," *Proc. of the 6th Int. Symp. on legal knowledge and legal reasoning systems*, Tokyo, Oct., 1992, pp. 32-54.

[7] K. Yokota, H. Tsuda, Y. Morita, S. Tojo, H. Yasukawa, "Specific features of a deductive object-oriented database language $\mathcal{QUIXOTE}$," *Proc. of the workshop on combining declarative and object-oriented databases*, ACM SIGMOD, Washington, D.C., May 29, 1993.