

The Vienna Sales Convention in a DOOD Language

Chie Takahashi

Japan Information Processing
Development Center(JIPDEC)
Kikaishinkou-Kaikan, 3-5-8,
Shibakoen, Minato-ku, Tokyo 105, JAPAN.
e-mail: j-takaha@icot.or.jp.

Kazumasa Yokota

Institute for New Generation
Computer Technology (ICOT)
21F., Mita-Kokusai Bldg., 1-4-28,
Mita, Minato-ku, Tokyo 108, JAPAN.
e-mail: kyokota@icot.or.jp.

Abstract

Legal reasoning is one application of large-scale knowledge information processing, where artificial intelligence, natural language processing, databases and other technologies are integrated. From the database point of view, legal reasoning requires access to vast data and knowledge sources such as written law and unwritten law. Because it is difficult for a person to deal with such amounts of legal data and knowledge, it is hoped that databases can support powerfully legal reasoning. So legal reasoning is the target for next generation databases.

In order to investigate whether or not the deductive object oriented database (DOOD) language/system *QUIXOTE* is effective in legal reasoning, we are both writing the *Vienna Sales Convention* (whose formal name is the *United Nations Convention on Contracts for the International Sale of Goods (CISG)*) in *QUIXOTE* and constructing the legal database on *QUIXOTE*.

In this paper, we show that *QUIXOTE* is suitable for representing legal data and knowledge.

1 Introduction

Legal reasoning is one application of large-scale knowledge information processing where artificial intelligence, natural language processing, databases and other technologies are expected to be integrated. From the database point of view, legal reasoning requires access to vast data and knowledge sources such as written law (for example, constitution, decrees, orders, ordinances, regulations and so on) and unwritten law (for example, customary law, case law, theories, social norms, industrial policies and so on). In particular, we have a great many precedents. Because it is difficult for a person to deal with such amounts of legal data and knowledge, it

is hoped that databases can manage them powerfully for legal reasoning. Although there are many legal databases where laws and precedents are stored in the form of natural language, it is said that they are not so useful and even that it is possible for a person to find correct legal data and knowledge in written and unwritten law faster than the databases.

In the FGCS (Fifth Generation Computer System) project, we have designed and developed *QUIXOTE*, a deductive object-oriented database (DOOD) language/system [4, 6, 7, 8, 9]. From a logic programming point of view, it is also thought of as an extended constraint logic programming language based on subsumption constraints. In addition, its queries and answers are extended to be able to deal with hypothetical reasoning and restricted abduction [2, 10]. As a result of these many features, *QUIXOTE* as a knowledge representation language plays an important role in knowledge information processing requiring a high capability of representation and query processing such as legal reasoning, genetic information processing and natural language understanding [3, 5, 8].

At present, we are constructing a legal database using the *Vienna Sales Convention* (whose formal name is the *United Nations Convention on Contracts for the International Sale of Goods (CISG)*) and using it to investigate how to represent legal knowledge, and how effective *QUIXOTE* is in legal reasoning. In this paper, we introduce a method of representing legal data and knowledge included in *CISG*.

Section 2 briefly introduces *CISG* and gives that part of *CISG* explained in this paper. Section 3 introduces a method of representing legal data and knowledge based on *CISG*. Section 4 summarizes our future works.

2 CISG Part II. FORMATION OF THE CONTRACT

CISG is the acronym for the *United Nations Convention on Contracts for the International Sale of Goods*, which has 101 articles and is also known as the *Vienna Sales Convention*. Domestic trade follows domestic law, while international trade observes this convention. It was adopted at a conference for diplomats in Vienna on April 11, 1980, and took effect on January 1, 1988.

We are constructing a legal database using *Part II. FORMATION OF THE CONTRACT*, which is the core of *CISG*. In this paper, we show the part of *CISG Part II* that is necessary to conclude a contract and gives rights to the parties concerned in a contract during the negotiations for concluding the contract.

In Section 2.1, we briefly explain a simplified legal model of the articles which are part of *CISG Part II* as shown in Appendix A. In Section 2.2, we list common questions about *CISG Part II*.

2.1 A Simplified Legal Model of the Articles

The following briefly explains a simplified legal model of the articles in Appendix A.

To conclude a contract, two parties are necessary. One party makes an *offer* to the other party. If the *offer* reaches the offeree, it becomes *effective*. The offeree may make a *counteroffer* to the *effective offer* or give an *acceptance* of the *effective offer* to the other party. If the *acceptance* reaches the addressee, it becomes *effective* and a *contract* is concluded.

Figure 1 shows the situation of concluding a contract. The words shown in figure 1 form the constituent activities in *CISG Part II*.

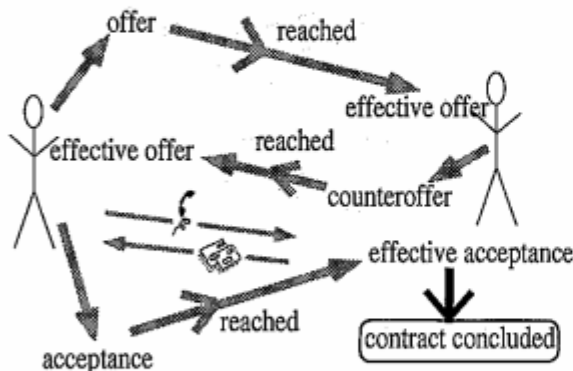


Figure 1: A Simplified Legal Model of the Articles

During a negotiation for concluding a contract, the parties concerned in the contract have the following rights:

- A party may withdraw his *offer* or his *acceptance* (Figure 2).
- A party may revoke his *effective offer* (Figure 3).

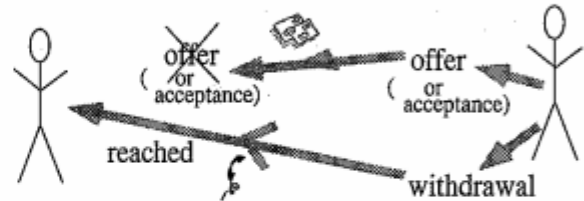


Figure 2: A right to withdraw an offer or an acceptance

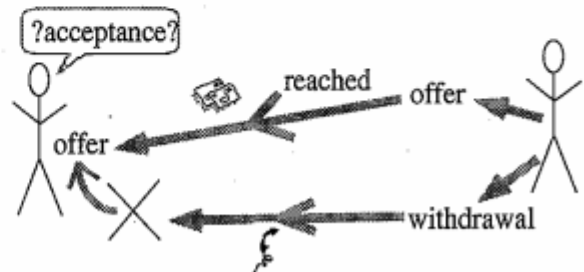


Figure 3: A right to revoke an effective offer

2.2 Common Questions about CISG Part II.

These are common questions about *CISG Part II*.

- "Does a proposal constitute an *offer*?"
- "Does an indication of intention to an *offer* constitute a *counteroffer*?"
- "Does an *offer* become *effective*?"
- "Does an indication of intention to an *offer* constitute an *acceptance*?"
- "Does an *acceptance* become *effective*?"
- "Is a *contract* concluded?"
- "May an *offer* be withdrawn?"
- "May an *effective offer* be revoked?"
- "May an *acceptance* be withdrawn?"

It is important that we can ask databases the above questions easily.

3 CISG Part II. in QUIXOTE

3.1 Objects and Modules in CISG Part II.

An article usually constitutes a legal concept when an incident meets some conditions imposed by the article. So at first we consider incidents in the real world and defined legal concepts in *CISG Part II*.

The following are incidents in the real world, defined legal concepts, and the articles that constitute the legal concepts in Appendix A.

1) Incidents in the real world.

- a party concerned in a contract (for example, Tom)
- the other party concerned in the contract (for example, Joe)
- a proposal (for example, Tom's proposal to Joe)
- some indications of intention of the one party during a negotiation for concluding the contract (for example, the number of Tom's indications is N)
- some indications of intention of the other party during the negotiation (for example, the number of Joe's indications is M)

2) Legal concepts and the articles which constitute them

- an offer(Article 14(1))
- a counteroffer for the offer, a counteroffer for the counteroffer for the offer, ... (Article 19(1))
- an offer may be withdrawn(Article 15(2))
- an effective offer(Article 15(1))
- an effective offer may be revoked(Article 16(1))
- an acceptance(Article 18(1))
- an acceptance may be withdrawn(Article 22)
- an effective acceptance(Article 18(2))
- a concluded contract(Article 23)
- an offer reaches, a counteroffer reaches, ..., an acceptance reaches, ... (Article 24)

The legal concepts in 2) can be divided into three: ones related to effectiveness, which need some information about time, others related to rights, and the others. *QUIXOTE* has *object terms* which represent objects or concepts and *modules* which classify data, knowledge or concepts. In order to represent the articles in Appendix A in *QUIXOTE*, we consider 4 *modules*: *module fact*, *def*, *effect_def* and *capable* added 1). We also consider *object terms* that represent the above incidents and the above legal concepts.

The 4 *modules* and the *object terms* are as follows:

(1) *module fact*:

Module fact has *object terms* that represent an incident in the real world.

- *tom* (He is a party concerned in the contract)
- *joe* (He is a party concerned in the contract)
- *proposal*[*offeror*=*tom*, *offeree*=*joe*]
- *indication*[*doer*=*joe*, *addressee*=*tom*, *order*=1],
...,
indication[*doer*=*joe*, *addressee*=*tom*, *order*= N]¹
- *indication*[*doer*=*tom*, *addressee*=*joe*, *order*=1],
...,
indication[*doer*=*tom*, *addressee*=*joe*, *order*= M]²

(2) *module def*:

Module def has *object terms* that represent a legal concept except effective ones.

- *offer*[*offeror*=*joe*, *offeree*=*tom*, *order*=1],
...,
offer[*offeror*=*joe*, *offeree*=*tom*, *order*= N]³
- *offer*[*offeror*=*tom*, *offeree*=*joe*, *order*=0],
...,
offer[*offeror*=*tom*, *offeree*=*joe*, *order*= M]⁴
- *acceptance*[*accepter*=*joe*, *offeror*=*tom*, *order*= N]
- *contract*[*accepter*=*joe*, *offeror*=*tom*, *order*= N]
- *reaches*[*doer*=*joe*, *addressee*=*tom*, *order*=1],
...,
reaches[*doer*=*joe*, *addressee*=*tom*, *order*= N],
reaches[*doer*=*tom*, *addressee*=*joe*, *order*=0],
...,
reaches[*doer*=*tom*, *addressee*=*joe*, *order*= M]

(3) *module effect_def*:

Module effect_def has *object terms* that represent a legal effective concept.

- *offer*[*offeror*=*joe*, *offeree*=*tom*, *order*=1],
...,
offer[*offeror*=*joe*, *offeree*=*tom*, *order*= N]

¹*indication*[*doer*=*joe*, *addressee*=*tom*, *order*= i] represents Joe's indication of intention to Tom.

indication[*doer*=*joe*, *addressee*=*tom*, *order*= $i + 1$]

represents the next indication to

indication[*doer*=*joe*, *addressee*=*tom*, *order*= i]($1 \leq i \leq N$)

²*indication*[*doer*=*tom*, *addressee*=*joe*, *order*= j]($1 \leq j \leq M$) represents Tom's indication of intention to Joe. They are also in order of time.

³*offer*[*offeror*=*joe*, *offeree*=*tom*, *order*= i] represents a counteroffer ($1 \leq i \leq N$).

⁴*offer*[*offeror*=*tom*, *offeree*=*joe*, *order*= j] represents a counteroffer ($1 \leq j \leq M$).

- $offer[offeror=tom, offeree=joe, order=0]$,
...,
 $offer[offeror=tom, offeree=joe, order=M]$
 - $acceptance[accepter=joe, offeror=tom, order=N]$
- (4) *module capable*:
Module capable has *object terms* that represent rights which the parties concerned in a contract have.
- $withdrawing_offer[offeror=joe, offeree=tom, order=K]$
 - $revoking_effective_offer[offeror=tom, offeree=joe, order=L]$
 - $withdrawing_acceptance[accepter=joe, offeror=tom, order=N]$

When we examine the relations between the above *object terms* and the articles in Appendix A, we notice that each article defines a correspondence between objects (Figure 4, Figure 5, and Figure 6).

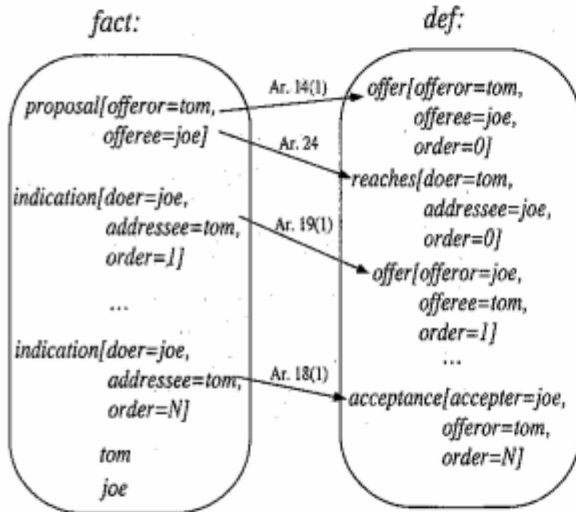


Figure 4: *Module fact* and *module def*

$def::offer[offeror=joe, offeree=tom, order=K]$ represents that Joe's K th indication of intention to Tom, which exists in the real world, constitutes an offer according to *CISG*. In the same way,

$effect_def::offer[offeror=joe, offeree=tom, order=K]$
 $def::reaches[doer=joe, addressee=tom, order=K]$
 $def::acceptance[accepter=joe, offeror=tom, order=K]$
 $def::contract[accepter=joe, offeror=tom, order=K]$

represent that Joe's K th indication of intention in the real world constitutes

- an effective offer
- a reached indication
- an acceptance
- a concluded contract

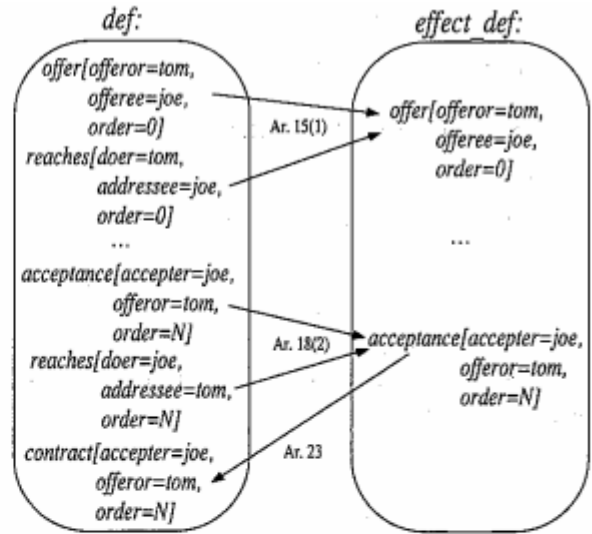


Figure 5: *Module def* and *module effect_def*

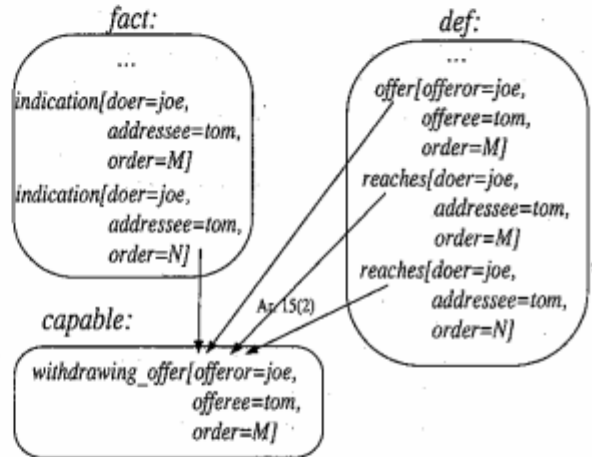


Figure 6: *Module capable* and other *modules*

respectively according to *CISG*.

$capable::withdrawing_offer[offeror=joe, offeree=tom, order=K]$ represents that Joe's K th offer may be withdrawn according to *CISG*. In the same way,

$capable::revoking_effective_offer[offeror=joe, offeree=tom, order=K]$
 $capable::withdrawing_acceptance[accepter=joe, offeror=tom, order=K]$

represent that

- Joe's K th effective offer may be revoked
- Joe's K th acceptance may be withdrawn

respectively according to *CISG*.

3.2 The Common Questions in *QUIXOTE*

The common questions introduced in Section 2.2 are represented in *QUIXOTE* as follows:

- “Does a proposal constitute an *offer*?”
?- *def:offer*[*offeror=tom, offeree=joe, order=0*].
- “Does an indication of intention to an *offer* constitute a (*counter-*) *offer*?”
?- *def:offer*[*offeror=A, offeree=B, order=K*].
((*A, B*)=(*tom, joe*)or(*joe, tom*), $1 \leq K \leq N, M$)
- “Does an *offer* become *effective*?”
?- *effect_def:offer*[*offeror=A, offeree=B, order=K*].
((*A, B*)=(*tom, joe*)or(*joe, tom*), $0 \leq K \leq N, M$)
- “Does an indication of intention to an *offer* constitute an *acceptance*?”
?- *def:acceptance*[*accepter=joe, offeror=tom, order=N*].
- “Does an *acceptance* become *effective*?”
?- *effect_def:acceptance*[*accepter=joe, offeror=tom, order=N*].
- “Is a *contract* concluded?”
?- *def:contract*[*accepter=joe, offeror=tom, order=N*].
- “May an *offer* be withdrawn?”
?- *capable::withdrawing_offer*[*offeror=A, offeree=B, order=N*]
((*A, B*)=(*tom, joe*)or(*joe, tom*), $0 \leq K \leq N, M$)
- “May an *effective offer* be revoked?”
?- *capable::revoking_effective_offer*
[*offeror=joe, offeree=tom, order=N*]
((*A, B*)=(*tom, joe*)or(*joe, tom*), $0 \leq K \leq N, M$)
- “May an *acceptance* be withdrawn?”
?- *capable::withdrawing_acceptance*
[*accepter=joe, offeror=tom, order=N*]
((*A, B*)=(*tom, joe*)or(*joe, tom*), $0 \leq K \leq N, M$)

3.3 Queries and Answers in *QUIXOTE*

One of the main features of data and knowledge in knowledge information processing, such as legal reasoning, is that the information is partial. That is to say, sufficient information is not necessarily given in the beginning. For example, a new case might not have all the important facts or new facts might be found later. So a query and

an answer are extended to be able to deal with partial information.

We introduce some questions and answers about *CISG Part II* in *QUIXOTE*.

We assume that some facts about a negotiation between Tom and Joe for concluding a contract are stored in the database.

1. “Does Tom’s proposal to Joe constitute an *offer*?”

This question is represented in *QUIXOTE* as follows:
?- *def:offer*[*offeror=tom, offeree=joe, order=0*].

We can get 4 answers with assumptions as follows:

```

IF fact:proposal[offeror=tom, offeree=joe].
    the_quantity_and_the_price  $\sqsubseteq$ 
        expressly_fixing_them
    fact:proposal[offeror=tom, offeree=joe].
        indicating_the_goods  $\cong$  yes
THEN YES
IF fact:proposal[offeror=tom, offeree=joe].
    the_quantity_and_the_price  $\sqsubseteq$ 
        implicitly_fixing_them
    fact:proposal[offeror=tom, offeree=joe].
        indicating_the_goods  $\cong$  yes
THEN YES
IF fact:proposal[offeror=tom, offeree=joe].
    the_quantity_and_the_price  $\sqsubseteq$ 
        making_provision_for_determining_them
    fact:proposal[offeror=tom, offeree=joe].
        indicating_the_goods  $\cong$  yes
THEN YES
IF fact:proposal[offeror=tom, offeree=joe].
    sufficiently_definite  $\cong$  yes
THEN YES

```

The first answer means that

“If the proposal expressly fixes the quantity and the price, and indicates the goods, then *YES*.”

Assumptions we get with answers are not part of the information stored in the database. In this way, queries clarify information missing from the database.

2. We assume that from the real proposal paper we find that the goods in the proposal are oranges, the quantity is 1000 and the price is 20 and we store this information in the following form:

```

fact::proposal[offeror=tom, offeree=joe]
/[goods="orange", quantity=1000, price=20].

```

In addition, we assume that a proposal indicating the goods is construed as fixing a value of the *goods label* in the above form, and a proposal expressly fixing the quantity and the price is construed as fixing values of the *quantity label* and the *price label* in the above form. If we want to ask the following question:

“Does the proposal constitute an *offer*?”
in this situation, it is represented in *QUIXOTE* by the following.

```
?-def:offer[offeror=tom, offeree=joe, order=0];;
    &program;;
fact::proposal[offeror=tom, offeree=joe]
    /[goods="orange", quantity=1000, price=20];;
fact::proposal[offeror=tom, offeree=joe]
    /[indicating_the_goods=yes] <-
    fact:proposal[offeror=tom, offeree=joe]
        /[goods → &string];;
fact::proposal[offeror=tom, offeree=joe]
    /[the_quantity_and_the_price →
        expressly_fixing_them] <-
    fact:proposal[offeror=tom, offeree=joe]
        /[quantity → &integer,
            price → &integer];;
```

As the above question, we can add a program to a conventional query in *QUIXOTE*. Added programs are dealt with as hypotheses.

3. “Does the *offer* become *effective*?”

This question is represented in *QUIXOTE* as follows:

```
?- effect_def:offer[offeror=tom,
    offeree=joe, order=0].
```

We assume that we know only that the proposal is delivered. We can get 4 answers with assumptions as follows:

```
IF fact:proposal[offeror=tom, offeree=joe].act
    ≅ delivery[place=personally]
THEN YES
IF fact:proposal[offeror=tom, offeree=joe].act
    ≅ delivery[place=business]
THEN YES
IF fact:proposal[offeror=tom, offeree=joe].act
    ≅ delivery[place=mailing_address]
THEN YES
```

The first answer means that

“If the proposal is delivered to Joe personally, then *YES*.”

4. We assume that Joe’s second indication of intention constitute a (*counter-*) *offer* and we want to withdraw the offer. We can ask the question:

“May the *offer* be withdrawn?”

represented in *QUIXOTE* as follows:

```
?- capable:withdrawing_offer[offeror=joe,
    offeree=tom, order=2].
```

We can get the following answer.

IF

```
fact:indication[order=3, doer=joe,
    addressee=tom].kind
    ≅ remove[object=indication[order=2,
        doer=joe, addressee=tom]]
def:reaches[order=3, doer=joe,
    addressee=tom].the_date_of_reaching
    ⊆5 def:reaches[order=2, doer=joe,
        addressee=tom].the_date_of_reaching
```

THEN YES

The answer means that

“If Joe’s third indication which removes Joe’s second indication and the third one reaches Tom before or at the same time as the second one, then *YES*.”

And if the condition is satisfied, we can actually delete Joe’s second indication by using an update rule in *QUIXOTE*[9].

5. We may want to know when Joe’s second indication reached Tom in order to examine whether or not the above condition is satisfied.

We can ask the question:

“What is the date when Joe’s second indication reached Tom?”

as follows:

```
?- def:reaches[order=2, doer=joe, addressee=tom]
    /[the_date_of_reaching=X].
```

For example, we can get the following answer:

```
X ≅ date[year=1994, month=11, day=20].
```

As the above queries and answers, we can deal with partial information, hypothetical reasoning and restricted abduction, which are important in legal reasoning, using *QUIXOTE*. In addition, we can know whether or not facts satisfy legal requirements for concluding a

⁵We use *subsumption constraint* instead of time constraint (written \leq_{time}) in this paper. We plan to improve *QUIXOTE* so that it can deal with time constraints. See Section 4.

contract, what facts are necessary to satisfy the legal requirements, and simulate, for example, a negotiation for concluding a contract when the person concerned in the contract exercises his rights.

3.4 Articles in *QUIXOTE*

In Appendix B, the articles in Appendix A are written in *QUIXOTE* using the *object terms* and the *modules* in Section 3.1. When we considered descriptions of the articles in *QUIXOTE*, we tried to make the descriptions represent the articles with regard to the legal model in Section 2.1. *QUIXOTE* has *attribute terms*, each of which represents an *object term* and its properties. Because an article usually connects an incident to a legal concept, we also tried to make the descriptions represent conditions in the articles as properties of *object terms*, that is, *attribute terms* in *module fact*, not in *module def* nor in *module effect.def*.

For example, article 19 (1) in Appendix A:

A reply to an offer which purports to be an acceptance but contains additions, limitations or other modifications is a rejection of the offer and constitutes a counteroffer.

is represented by the rule in Appendix B:

```
def::offer[offeror=A, offeree=B, order=N] ←
  fact:indication[doer=A, addressee=B, order=N]
  /{act → indication_of_intention,
    intention=assent,
    containing_modifications=yes,
    kind=reply[object=
      indication[doer=B, addressee=A, order=M]]};
effect.def:offer[offeror=B, offeree=A, order=M];;
```

The rule means that

When A's Nth indication of assent which is a reply to B's Mth indication and contains modifications exists in the real world, and the B's Mth indication constitutes an effective offer, the Nth indication constitutes an (counter)offer.

As the other example, article 15 (2) in Appendix A:

An offer, even if it is irrevocable, may be withdrawn if the withdrawal reaches the offeree before or at the same time as the offer.

is represented by the rule in Appendix B:

```
capable::withdrawing_offer[offeror=A, offeree=B,
  order=N] ←
  fact:indication[doer=A, addressee=B, order=M]
  /{kind=remove[object=
```

```
  indication[doer=A, addressee=B, order=N]]};
def:offer[offeror=A, offeree=B, order=N];
def:reaches[doer=A, addressee=B, order=N]
  /{the_date_of_reaching=S0};
def:reaches[doer=A, addressee=B, order=M]
  /{the_date_of_reaching=S1}
||{S1 ⊆6S0};;
```

The rule means that

When A's Mth indication of removing A's Nth indication of intention exists in the real world, the Nth indication constitutes an offer, the date of reaching of the offer is S0, the date of reaching of the Mth indication is S1, and S1 is earlier than S0, the offer may be withdrawn.

4 Concluding Remarks

Although legal reasoning is one of the most attractive applications for next generation databases, there have not been many studies involving legal databases. In this paper, we represent legal data and knowledge in the framework of a DOOD. Our contributions in this paper can be summarized as follows:

1. We show that *QUIXOTE*, as a DOOD language /system, is effective for representing legal data and knowledge properly.
2. We show that its query, which is able to deal with partial information, hypothetical reasoning and restricted abduction, is useful in legal reasoning, and a legal database on *QUIXOTE* is more powerful than a conventional one.

To create a bigger legal database, we plan the following experiments and extensions:

(1) Control of generating assumptions

Although *QUIXOTE* gives only properties of objects as assumption with answers and we can make each rule not generate assumptions in query processing, we sometimes get a lots of assumptions. We are considering how to get only the important assumptions or only those assumptions which we want to know.

⁶We use *subsumption constraint* instead of time constraint (written \leq_{time}) in this paper. We plan to improve *QUIXOTE* so that it can deal with time constraints. See Section 4.

(2) time constraints

We are trying to improve *QUIXOTE* so that it can use external functions. For example, if *QUIXOTE* can use an external function that evaluates time constraints (written \leq_{time} and $<_{time}$, for example, comparing two dates in order to find out which one is earlier and counting the number of days), we can describe a *module effect[time=T]* in *QUIXOTE* as follows:

$$\begin{aligned} \text{effect[time=T]}::X \Leftarrow \\ \text{effect_def:} X / [\text{the_date_of_beginning}=S, \\ \text{the_date_of_terminated}=E] \\ \|\{S \leq_{time} T, T <_{time} E\}. \end{aligned}$$

The *module effect[time=T]* contains the *object terms* effective at time T .

(3) Negation

QUIXOTE has *NAF* (Negation as Failure) and *disequation* constraints to deal with negative information. We are investigating whether or not the above 2 features are enough to represent the *Vienna Sales Convention* in *QUIXOTE* and whether it is better to extend *QUIXOTE* to be able to deal with non-monotonic reasoning for representing law naturally.

(4) Helios

Although ‘reasonable’ is used in many articles of CISG, the meaning of the word is not defined in CISG. The reason is that the court wants to clarify the meaning without the influence of social and period changes. A powerful legal database system thus has a function that adopts human judgement.

Helios which is a heterogeneous, distributed, cooperative problem-solving system, is being studied and developed at ICOT [11]. *Helios* can define any database, constraint solver, application program, or even a person, as an agent, and can solve problems by cooperation among agents.

Using *Helios* with *QUIXOTE* as a problem solver, we can create a powerful legal database system for CISG on *Helios* which adopts human judgement.

We will expand the legal database using CISG to a very large database/knowledge base as we enhance *QUIXOTE* using *Helios*.

Acknowledgments

The authors wish to thank all the members of the *QUIXOTE* project for their valuable advice and comments.

Appendix A: Part of CISG Part II.

Article 14

- (1) A proposal for concluding a contract addressed to one or more specific persons constitutes an offer if it is sufficiently definite and indicates the intention of the offeror to be bound in case of acceptance. A proposal is sufficiently definite if it indicates the goods and expressly or implicitly fixes or makes provision for determining the quantity and the price.

Article 15

- (1) An offer becomes effective when it reaches the offeree.
- (2) An offer, even if it is irrevocable, may be withdrawn if the withdrawal reaches the offeree before or at the same time as the offer.

Article 16

- (1) Until a contract is concluded an offer may be revoked if the revocation reaches the offeree before he has dispatched an acceptance.

Article 18

- (1) A statement made by or other conduct of the offeree indicating assent to an offer is an acceptance. Silence or inactivity does not in itself amount to acceptance.
- (2) An acceptance of an offer becomes effective at the moment the indication of assent reaches the offeror. An acceptance is not effective if the indication of assent does not reach the offeror within the time he has fixed or, if no time is fixed, within a reasonable time, due account being taken of the circumstances of the transaction, including the rapidity of the means of communication employed by the offeror. An oral offer must be accepted immediately unless the circumstances indicate otherwise.

Article 19

- (1) A reply to an offer which purports to be an acceptance but contains additions, limitations or other modifications is a rejection of the offer and constitutes a counteroffer.

Article 22

An acceptance may be withdrawn if the withdrawal reaches the offeror before or at the same time as the acceptance would have become effective.

Article 23

A contract is concluded at the moment when an acceptance of an offer becomes effective in accordance with the provisions of this Convention.

Article 24

For the purposes of this Part of the Convention, an offer, declaration of acceptance or any other indication of intention "reaches" the addressee when it is made orally to him or delivered by any other means to him personally, to his place of business or mailing address or, if he does not have a place of business or mailing address, to his habitual residence.

Appendix B: Articles in *QUIXOTE*

Article 14(1)

```
def::offer[offeror=A, offeree=B, order=0] ←
  fact:proposal[offeror=A, offeree=B]
    /[having_a_purpose_to_conclude_a_contract=yes,
      addressed_to_one_or_more_specific_persons=yes,
      sufficiently_definite=yes,
      indicating_the_intention_of_the_offeror_
        to_be_bound_in_case_of_acceptance=yes];;
fact::proposal[offeror=A, offeree=B]
  /[sufficiently_definite=yes] ←
  fact:proposal[offeror=A, offeree=B]
    /[indicating_the_goods=yes,
      the_quantity_and_the_price → expressly_fixing_them];;
fact::proposal[offeror=A, offeree=B]
  /[sufficiently_definite=yes] ←
  fact:proposal[offeror=A, offeree=B]
    /[indicating_the_goods=yes,
      the_quantity_and_the_price → implicitly_fixing_them];;
fact::proposal[offeror=A, offeree=B]
  /[sufficiently_definite=yes] ←
  fact:proposal[offeror=A, offeree=B]
    /[indicating_the_goods=yes,
      the_quantity_and_the_price →
        making_provision_for_determining_them];;
```

Article 15(1)

```
effect_def::offer[offeror=A, offeree=B, order=N]
  /[the_date_of_becoming_effective=S] ←
  def:offer[offeror=A, offeree=B, order=N];
  def:reaches[doer=A, addressee=B, order=N]
    /[the_date_of_reaching=S];;
```

Article 15(2)

```
capable::withdrawing_offer[offeror=A, offeree=B, order=N] ←
  fact:indication[doer=A, addressee=B, order=M]
    /[kind=remove[object=
      indication[doer=A, addressee=B, order=N]]];;
def:offer[offeror=A, offeree=B, order=N];
def:reaches[doer=A, addressee=B, order=N]
  /[the_date_of_reaching=S0];
def:reaches[doer=A, addressee=B, order=M]
  /[the_date_of_reaching=S1]
  ||{S1 ⊆ 7S0};;
```

Article 16(1)

```
capable::revoking_offer[offeror=A, offeree=B, order=N] ←
  fact:indication[doer=A, addressee=B, order=M]
    /[kind=remove[object=
      indication[doer=A, addressee=B, order=N]]];;
def:offer[offeror=A, offeree=B, order=N];
def:reaches[doer=A, addressee=B, order=N]
  /[the_date_of_reaching=S0];
def:reaches[doer=A, addressee=B, order=M]
  /[the_date_of_reaching=S1];
def:acceptance[accepter=B, offeror=A, order=K]
  /[object=
    indication[doer=A, addressee=B, order=N]];
fact:indication[doer=B, addressee=A, order=K]
  /[the_date_of_dispatching=S2]
  ||{S0 ⊆ S1, S0 ≠ S1 8, S0 ⊆ S2, S1 ⊆ S2, S1 ≠ S2} 9;;
```

Article 18(1)

```
def::acceptance[accepter=A, offeror=B, order=N]
  /[object=
    indication[doer=B, addressee=A, order=M]] ←
  fact:indication[doer=A, addressee=B, order=N]
  /[act → indication_of_intention,
    intention=assent,
    kind=reply[object=
      indication[doer=B, addressee=A, order=M]],
    containing_modifications=no] 10;
effect_def:offer[offeror=B, offeree=A, order=M];;
```

Article 18(2)

⁷We use *subsumption constraint* instead of time constraint (written \leq_{time}) in this paper. We plan to improve *QUIXOTE* so that it can deal with time constraints. See Section 4.

⁸As the result of debugging the legal database, $\{S0 \subseteq S1, S0 \neq S1\}$ is added. For details of debugging legal databases, see [5].

⁹We use *subsumption constraint* instead of time constraint (written \leq_{time}) in this paper. These constraints are represented as $\{S0 <_{time} S1, S0 \leq_{time} S2, S1 <_{time} S2\}$ by using time constraint.

¹⁰As the result of debugging the legal database, $\{S0 \subseteq S1, S0 \neq S1\}$ is added.

effect_def::acceptance[offeror=A, acceptor=B, order=N]
 /[the_date_of_becoming_effective=S] \Leftarrow
 def::acceptance[acceptor=B, offeror=A, order=N];
 def::reaches[doer=B, addressee=A, order=N]
 /[kind \rightarrow within_a/the_time, the_date_of_reaching=S];;

Article 19(1)

def::offer[offeror=A, offeree=B, order=N] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act \rightarrow indication_of_intention,
 intention=assent,
 containing_modifications=yes,
 kind=reply(object=
 indication[doer=B, addressee=A, order=M])];
 effect_def::offer[offeror=B, offeree=A, order=M];;

Article 22

capable::withdrawing_acceptance
 [acceptor=A, offeror=B, order=N] \Leftarrow
 fact::indication[doer=A, addressee=B, order=M]
 /[kind=remove(object=
 indication[doer=A, addressee=B, order=N])];
 def::acceptance[acceptor=A, offeror=B, order=N];
 def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=S0];
 def::reaches[doer=A, addressee=B, order=M]
 /[the_date_of_reaching=S1]
 ||{S1 \subseteq ¹¹S0};;

Article 23

def::contract[offeror=A, acceptor=B, order=N] \Leftarrow
 effect_def::acceptance[offeror=A, acceptor=B, order=N];;

Article 24

def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=T] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act=oral_conduct, the_date_of_oral_conduct=T];;
 def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=T] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act=delivery(place=personally),
 the_date_of_delivery=T];;
 def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=T] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act=delivery(place=business),
 the_date_of_delivery=T];;

¹¹We use *subsumption constraint* instead of time constraint (written \leq_{time}) in this paper.

fact:B/[having_a_place_of_business_or_
 mailing_address=yes]
 ||{B \subseteq the_party_concerned};;
 def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=T] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act=delivery(place=mailing_address),
 the_date_of_delivery=T];
 fact:B/[having_a_place_of_business_or_
 mailing_address=yes]
 ||{B \subseteq the_party_concerned};;
 def::reaches[doer=A, addressee=B, order=N]
 /[the_date_of_reaching=T] \Leftarrow
 fact::indication[doer=A, addressee=B, order=N]
 /[act=delivery(place=habitual_residence),
 the_date_of_delivery=T];
 fact:B/[having_a_place_of_business_or_
 mailing_address=no]
 ||{B \subseteq the_party_concerned};;

References

- [1] S. Niibori, "The Uniform Law on International Sales — Vienna Sales Convention and Trade Contract —", Doubunkan Publishing Company, 1991. (in Japanese)
- [2] T. Nishioka, R. Ojima, H. Tsuda and K. Yokota, "Procedural Semantics of a DOOD Programming Language *QUIXOTE*", *Joint Workshop of IPSJ SIGDSS and IEICE SIGDE (EDWIN)*, Nagasaki, July 21-23, 1993. (in Japanese)
- [3] T. Nishioka, K. Yokota, C. Takahashi and S. Tojo, "Constructing a Legal Knowledge-base with Partial Information", *Proc. ECAI'94 Workshop on Artificial Normative Reasoning*, Aug.8, 1994.
- [4] C. Takahashi, Y. Morita, T. Nishioka and H. Tsuda, "Features and Implementation of a Deductive Object-Oriented Database System *QUIXOTE*", *Joint Workshop of IPSJ SIGDSS and IEICE SIGDE (EDWIN)*, Nagasaki, July 21-23, 1993. (in Japanese)
- [5] Chie Takahashi and Kazumasa Yokota, "Constructing a Legal Database on *QUIXOTE*", *Proc. the Sixth Australasian Database Conference (ADC'95)*, Adelaide, Australia, Jan. 30,31, 1995.
- [6] H. Tsuda and K. Yokota, "Knowledge Representation Language *QUIXOTE*", *Proc. Int. Conf. on FGCS, ICOT*, Tokyo, Dec. 13-14, 1994.

- [7] H. Yasukawa, H. Tsuda, and K. Yokota. "Objects, Properties, and Modules in *QUIXOTE*", *Proc. Int. Conf. on FGCS, ICOT, Tokyo, June 1-5, 1992*.
- [8] K.Yokota and H.Yasukawa, "Towards an Integrated Knowledge-Base Management System — Overview of R&D on Databases and Knowledge-Bases in the FGCS Project", *Proc. Int. Conf. on FGCS, ICOT, Tokyo, June 1-5, 1992*.
- [9] K.Yokota, H.Tsuda and Y.Morita, "Specific Features of a Deductive Object-Oriented Database Language *QUIXOTE*", *Proc. Workshop on Combining Declarative and Object-Oriented Databases, (ACM SIGMOD'93 Workshop), Washington DC, May 29, 1993*.
- [10] K.Yokota, T.Nishioka, H.Tsuda and S.Tojo, "Query Processing for Partial Information Database in *QUIXOTE*", *Proc. Int. Conf. on Tools with Artificial Intelligence, New Orleans, Nov 6-9, 1994*.
- [11] K. Yokota and A. Aiba, "A New Framework of Very Large Knowledge Bases", *Knowledge Building and Knowledge Sharing*, eds K. Fuchi and T. Yokoi, Ohmsha and IOS Press, 1994.