

# 複合商品を扱う電子商取引のための動的制約代数処理系の構築と応用

檜崎潤\* 岩井原瑞穂\*\* 上林弥彦\*\*

## 要約

電子商取引の交渉では、価格や数量以外にも割引条件や納品条件、可能な商品の組合せなど多様な制約条件が存在する。このような商取引に関する制約を動的制約として定式化し、動的制約のデータベースに対する質問言語として動的制約代数を提案している。本稿では動的制約代数の処理系の構築について述べる、その評価を行う。本稿ではPC販売サイトのデータに対してその処理系を適用してみる。この処理系では利用者が選択した組合せのプル条件による動的制約をもとに、それを充足可能な組合せを含む機種を検索することができる。最後に動的制約代数を電子商取引に利用するための利用者インタフェースについて述べる。

## 第1章 研究開発の背景及び目的

既存の電子商取引システムにおいて単一商品を価格などの属性値で検索する場合は、従来の情報検索システムを用いることができる。しかし、複数の商品を組合せる複合商品を購入する場合の商品の組合せによる検索方法では組合せの制約による検索が必要である。そのため本論文では動的制約代数を用いた複合商品の検索手法およびそのための利用者インタフェースの構築について述べる。

複合商品の制約の例として含意の制約がある：「もし  $x = a$  ならば  $y = b$  である」。一つの製品で現れる例は、ある Video Card ( $x = a$  によって表される) はある Operating System ( $y = b$  によって表される) を必要としているなどである。

電子商取引の制約を形式的に表すために、動的制約[1]の概念を用いる。論文[1]で提案した、動的制約の様々な属性を質問するための特別の演算子を持つ動的制約代数(DCA)により、電子商取引の動的制約データベースの検索が可能になる。

本論文ではこれらの方法を評価するために実際の電子商取引サイトにある製品の構成情報を動的制約に変換し、動的制約が電子商取引の製品の制約を表せることを確認する。

DCA と動的制約データベースを使用することにより以下の新しい検索機能を提供することができる。

- 現在のオンラインサイトや自然言語を基にした注文の記述に代わり、多様な買い手と売り手の条件の知的な照合ができる。
- たとえ買い手の購買計画が志望する値が不完全な情報を含んでいたとしても、マッチする可能性のある売り手の制約を求めることができる。
- 買い手と売り手はスキーマの制限がなく多様な動的制約を記述可能であり、そのような多様な動的制約に対して質問を実行できる。

以降、2章では、動的制約と動的制約代数(DCA)を定義し、DCA 質問の例を示す。3章では、DCA の充足可能性問題を示す。4章ではインターネットの販売サイトから抽出した動的制約の実際の例を示す。

5章は利用者インタフェースについて述べる。6章は結論である。

## 第2章 動的制約

### 2.1. 制約関係

| Item (品目) | Model (型) | Price (価格) | Condition(条件節)              |
|-----------|-----------|------------|-----------------------------|
| MPU       | $m_1$     | $p_1$      | $(m_1=500 \quad m_1=750)$   |
|           |           |            | $(m_1=500 \quad p_1=249)$   |
|           |           |            | $(m_1=750 \quad p_1=399)$   |
| RDRAM     | $m_2$     | $p_2$      | $m_2=128MB \quad m_2=256MB$ |
| SDRAM     | 128MB     | 100        | T                           |
| Fan       | Twin      | 60         | $m_1 = 750$                 |

図1：Product の制約関係

| Item (品目) | Model (型) | Condition(条件節) |
|-----------|-----------|----------------|
| MPU       | 750       | T              |
| $i_1$     | $n_1$     | $i_1 = RDRAM$  |
|           |           | $i_1 = SDRAM$  |

図2：OrderPlan の制約関係

制約関係と動的制約代数の形式的な定義を示す前に図1と図2で例を示す。

制約関係のタプルは属性の値として定数か制約変数を持つ。それぞれのタプルは制約式を持つ。もしタプルの式が真ならば、タプルの表す情報は妥当であるとみなす。もし式の中にあるいくつかの変数に値がなかったら、タプルの妥当性は不明とする。もし式が偽ならば、そのタプルは関係に依存しないとする。

図1の関係は売り手の商品のリストを示す。制約式がそれぞれのタプルで真になったときその商品と価格の組合せは正しいとみなす。変数はイタリック体のアルファベットと数字の添え字で示している。最初のタプルの MPU は型、価格にそれぞれ  $m_1$ ,  $p_1$

といった変数を持つ。タプルの条件は、 $m_1$ の値は500か750であること、もし $m_1$ が500であれば $p_1$ は249であること、そしてもし $m_1$ が750であれば $p_1$ は399であることが含まれている。

二番目のタプルである品目 RDRAM は変数 $m_2$ を型に持ち、 $m_2$ は128MBか256MBのどちらかの値を持つという制約を受ける。RDRAMの価格の値は変数 $p_2$ であり、 $p_2$ にはなんの制約も与えられていない。このケースでは、価格は与えられていない、つまり売り手がRDRAMの価格を明らかにしていないと解釈する。三番目のタプルの Fan-Twin は $m_1 = 750$ という制約を持つ。変数 $m_1$ はMPUの型であり、Fan-TwinはMPU-750と一緒にだけ選ばれることができるかと解釈する。

図2の OrderPlan の制約関係は買い手が買おうとしている品目のリストである。しかし、買い手は最終的な決定はしておらず、いくつかの値は不完全である。

制約関係は与えられていない値、候補値、妥当な値の組合せといった電子商取引の交渉のための様々な形の動的制約を表すのに適している。

## 2.2. 制約組の定義

制約式は、制約変数である $x_i$ 、比較演算子 $\{=, <, >, \leq, \geq\}$ である $\alpha_i$ 、定数または制約変数の $d_i$ を (連言)、(選言)、 $\neg$  (否定)、 $\subseteq$  (包含)で接続された項 $(x_i \alpha_i d_i)$ の式である( $i = 1, \dots, m$ )。

$T$ と $F$ をそれぞれブールの真と偽の値とする。任意の定数代入で $(f)$ と $(g)$ が同じブール値を返す場合に限り、制約式 $f$ と $g$ は等価であるとし、 $f \equiv g$ によって示す。もし $f \equiv F$ ならば $f$ は充足可能といい、そうでなければ $f$ は充足不能という。

**定義 1** 制約タプルは以下のように表される： $[a_1, \dots, a_k] /$  ,ここで $[a_1, \dots, a_k]$ はあるスキーマのタプルであり、 $a_i$ は定数か制約変数のどちらかである。は制約式である。

## 2.3. 動的制約代数

動的制約代数(DCA) [1]について説明する。買い手と売り手の多様な制約の集合について、照合したり性質を調べたりするのにDCAは適している。

集合和( $r \cup s$ )、直積( $r \times s$ )、差集合( $r - s$ )、選択( $\sigma_\rho(r)$ )、射影( $\pi_s(r)$ )、変数への代入( $S(r)$ )からなる制約関係代数を定義することができるが、紙面の都合上、集合和と選択の定義のみ示す。

### 定義 2

$r$ と $s$ はそれぞれ属性 $k$ と $m$ の制約関係である。

$$r \cup s \stackrel{\text{def}}{=} \{t / \mid t / \in r \text{ or } t / \in s\},$$

ここで $r$ と $s$ は同じスキーマを持つ( $k = m$ )。

$$\sigma_\rho(r) \stackrel{\text{def}}{=} \{[a_1, \dots, a_k] / (\rho') \mid [a_1, \dots, a_k] / \in r, \rho' \text{ is obtained by replacing each attribute } A_i \text{ in } \rho \text{ with corresponding element } a_i, \rho' \neq F\}$$

## 例 1 OrderPlan と Product を結合する。

この質問は品目と型の属性値によって二つの関係を結合する。これにより買い手と売り手の制約の同じ品目ごとに論理積を取る。結合は積( $\times$ )と選択( $\sigma$ )によって表現できる：

$$PO_1 = \sigma_{Item=Item\#2 \wedge Model=Model\#2} (Product \times OrderPlan)$$

ここで、右側にある Item と Model の「#2」は、積の引数の2つの関係のうち2番目の属性であることを示す。結果を図3に示す。買い手はRDRAMとSDRAMのどちらかを選択していないが、結合はそのような不完全な情報でも計算できる。

| Item  | Model | Price | Condition                                  |
|-------|-------|-------|--|
| MPU   | $m_1$ | $p_1$ | $m_1=750$ $p_1=399$                        |
| $i_1$ | $m_2$ | $p_2$ | $i_1=RDRAM$<br>( $m_2=128MB$ $m_2=256MB$ ) |
| $i_1$ | $n_1$ | 100   | ( $m_2=n_1$ )<br>$i_1=SDRAM$ $n_1=128MB$   |

図3 : Join of OrderPlan and Product

## 2.4. 動的制約演算子

次に動的制約演算子 $B$ 、 $C$ 、 $D$ を定義する。これらはスキーマを持たない動的制約の変数に限量子を与えて充足可能性を調べるものであり、電子商取引の交渉では売買条件の性質を検査するのに有用である。

### 定義 3 (Boolean) $B_X(r)$

$X \in \{T, M\}$ とする。 $B_X(r)$ は(1)  $X = T$ ならば $T$ 、(2)  $X = M$ ならば $\neq F$ であり $\neq T$ となるタプル $t /$ の集合である。

これは OrderPlan の中で買い手が注文する意志が確定的である品目を見つけるものであり、以下の質問で表される：

$$B_T(\text{OrderPlan})$$

結果はMPU-750のタプル一つからなる。 $i_1$ を持つタプルは答えではない。

### 定義 4 (Cardinality) $C_{|A|ak}(r)$

$A$  を  $r$  の属性とし、 $\alpha$  を比較演算子  $\{<, =, >\}$  とし、 $k$  を負でない整数か無限大とする。 $t/$  を  $r$  にあるタプルとする。 $S$  を  $( ) = T$  となる定数代入の集合とする。 $|A|$  を  $S$  の置換によって  $A$  に割り当てられた、異なった値の数とする。そして  $C_{|A|\alpha k}(r)$  の解は制約  $|A| \alpha k$  を充足するとするタプル  $t/$  の集合である。

$|A| < \text{inf}$  により  $|A|$  が有限の数であるとする。

**例 3** : 買い手が複数の候補を示している item を見つける。

この質問は品目の値が変数で、可能性のある値の集合が制約で定義されているタプルを見つける。

この質問は cardinality 演算で表現できる :

$$C_{1 < |Item| < \text{inf}}(\text{OrderPlan})$$

図 2 の 2 番目のタプルは品目の値が決まっていない。そしてその候補は RDRAM と SDRAM である。このタプルのみ関係が結果である。

さらに制約変数の依存関係について検索する Dependency 演算があるが省略する。

### 第 3 章 質問処理

#### 3.1. 命題論理による充足可能性への帰着

この章では DCA の充足可能性問題の概要を示す。以下で、定理と同等のシミュレートをする命題論理式を構成する。活動領域  $\text{adom}(\ )$  の代わりに以下の関数  $\text{adomq}(\ , x)$  を使う。それはそれぞれの  $x$  で可能性のある値の集合を算出し、それにより命題論理式の大きさを減少させる。

1.  $\text{adomq}(\ , x) = 0$
2. もし  $c$  が定数で、 $x = "c"$  または  $x = "c"$  となる項があれば、 $\text{adomq}(\ , x)$  に  $c$  を加える。
3. 制約変数  $x$  と  $y$  の組において、もし  $x = y$  または  $x \neq y$  となる項があれば、結合  $U = \text{adomq}(\ , x) \cap \text{adomq}(\ , y)$  をとり、 $\text{adomq}(\ , x) = U$ 、 $\text{adomq}(\ , y) = U$  とする。
4. あらゆる  $x$  において、 $\text{adomq}(\ , x)$  に加えられる要素が無くなるまで上の 2 と 3 を繰り返す。

#### 3.2. 質問処理アーキテクチャ

DCA の質問処理システムについて述べる。入力するデータベースはそれぞれのタプルが XML の形式となっており、データベース構文解析部でデータベースを構成要素に分析する。動的制約はタプルの要素として蓄えられる。質問は入力された質問ファイルを構文解析し、高位の最適化を行う。さらに動的制約の解析を行い、中間処理をはさんで、それらを制約

変換部に渡す。

制約変換部は動的制約を文献[1]に示した方法で命題論理式に変換する。変換された命題論理式を二分決定グラフ(BDD)処理パッケージに送り、そこで充足可能性/充足不可能性を演算し、その結果を質問処理部に送る。質問処理部ではその結果に基づいて解となるタプルを出力する。

### 第 4 章 実験

DCA 処理システムを用いた買い手と売り手の制約を照合させ検査することからなる電子商取引の交渉支援の実験を行う。

ベンチマークとして以下のデータを用いる :

1. 売り手の製品のデータとして PC 販売サイトからパソコンの価格構成表を抽出した。その構成表を制約に変換した。詳細は以下で述べる。
2. 買い手の要求として、動的制約の集合を与えた。

#### 4.1. 抽出した動的制約

表 1 に製品データの規模をまとめている。一つの製品はいくつかの品目から構成されている。RAID Card のようないくつかの品目は二三の製品にのみ現れるが、Processor や Memory といった品目は全ての製品に共通である。一つの品目の制約は制約パターンの論理積であり、一つの製品の動的制約は品目の動的制約の論理積である。

|                         |            |
|-------------------------|------------|
| 製品の数                    | 60         |
| 品目の数                    | 1173       |
| 製品ごとの品目の数<br>(最大/平均/最小) | 26/19.55/9 |
| 変数の数                    | 3519       |
| 定数の数                    | 6060       |
| 動的制約の数                  | 13788      |

表 1 : サイトから抽出した動的制約の特徴

サイトから抽出した制約を次の制約パターンに分類することができた。

- (選択)  $(x = c_1 \vee x = c_2 \vee \dots \vee x = c_k)$  選択制約同じ変数  $x$  の等式の和である。制約は  $x$  の可能性のある値を  $c_1, \dots, c_k$  に制限する。
- (組合せ)  $((x = c_1 \wedge y = d_1) \vee \dots \vee (x = c_k \wedge y = d_k))$  二つ(またはそれ以上)の変数の可能性のある組合せを記述している。
- (含意)  $(x = c \rightarrow y = b)$  もし  $x = c$  ならば  $y = b$  であるという規則を表す。
- (その他) 上のパターンではない制約

これらの動的制約で表現できない制約が存在したが、それは売り手が与えた選択条件ではなく、買い手が 500 人を超える会社であるような買い手の資格に関する制約である。これらを除いて条件を動的制約で

表現することができた。

次に、買い手の制約からなる関係 OrderPlan を与える。制約の例として、買い手は 128MB または 256MB の Memory を必要としている(選択)。買い手は IOMEGA drive か Tape drive を買おうとしている(組合せ)。OrderPlan に次のパターンも用いる：

- (定数) 属性の値に定数を用いる：買い手の(決定された)選択を表す。
- (否定) ( $x \neq c$ ) 20GB の Hard Drive を選択しないだろうといった買い手が選択しない型を表すのに用いる。
- (値無し) 属性に定数も変数も与えない。空の値を存在しないものとみなす。
- (制約無し) 属性の値に変数  $x$  が与えられているが、 $x$  に制約は与えられていない。品目に制限が無いとき、OrderPlan を制約無しのパターンで埋める。

#### 4.2. 性能評価

次に 4 個のベンチマーク質問を通じて DCA 質問処理システムの性能について議論する。それぞれの質問の実行時間を測り、また質問結果の表現サイズも測った。

次の質問 T1-T4 は買い手と売り手の制約を照合し、照合した制約の状態を質問するという基本的タスクから成っている。以下では、関係 Product は図 1 のスキーマと売り手から抽出した制約を持っている。関係 OrderPlan は図 2 のスキーマと買い手の制約を含んでいる。

T1 (join)

$$Match = \sigma_{Model\#1=Model\#2}$$

( $\sigma_{Item\#1=Item\#2} (Product \times OrderPlan)$ )

それぞれの品目で、買い手と売り手の制約を同じ型で結合させる。その結果の Match は品目の制約を満たすものとなる。

T2 (Boolean)

$$P_5 = B_M (CMatch)$$

制約がまだ真でない品目を求める。この質問はすでに制約を満足しているタプルを取り除くのに用いられる。

T3 (cardinality(>1))

$$P_6 = C_{Model\#1} (\sigma_{pid=pid\#2}$$

( $\sigma_{Item="OperatingSystem"} (P') \times Product$ ))

この質問は複数の Operating System を利用できる物を見つけるためのものである。

T4 (cardinality(=1))

$$P_7 = \sigma_{Item="OperatingSystem"} (P')$$

$\sigma_{Item="Processor"} \quad \sigma_{Item="Warranty"}$

$$P_7 = C_{Price=1} (\sigma_{pid=pid\#2} (CMatch \times P_7))$$

この質問は三つの品目で価格が一つに決まるものを見つける。

上の T1 から T4 の質問は Sun Enterprise 420R で実行した。表 2 は実行時間(3 回実行した平均)と結果タプルの数、タプルの制約を BDD で表現したサイズ、解の数(ここでは結果の制約を満たすすべての定数タプルの個数)を示している。

#### 4.3. 検討

表 2 の結果の検討をしてみる。それぞれの質問は現実的な実行時間内で計算されている。

解のサイズでは、実際の解の数は質問 T2-T4 で  $2^{32}$  を超えている。しかし解集合は制約式による表現および BDD による中間表現は十分小さなサイズであった。定数タプルによる通常の関係では解を表現することは実用的でないと考えられる。

買い手の制約 OrderPlan と売り手の制約 Product は未指定や候補の値としての不完全な情報が伴うが、DCA 質問では制約式の演算により可能な解を計算し、あるいは解の存在状況を調べることができる。

| 質問 | 説明               | 解の製品・品目を表現する制約タプルの数 | 時間(秒) | BDD ノード | 解を表現する定数タプルの数 |
|----|------------------|---------------------|-------|---------|---------------|
| T1 | Join             | 1001                | 24.73 | 38588   | 5970          |
| T2 | Bool             | 56                  | 27.67 | 124685  | $> 2^{32}$    |
| T3 | Cardinality (>1) | 45                  | 23.27 | 143365  | $> 2^{32}$    |
| T4 | Cardinality (=1) | 15                  | 75.13 | 8343    | $> 2^{32}$    |

表 2 : 質問の結果

## 第 5 章 利用者インタフェース

この章では動的制約代数を電子商取引に利用するた

めの利用者インタフェースについて述べる。

### 5.1. システム全体図

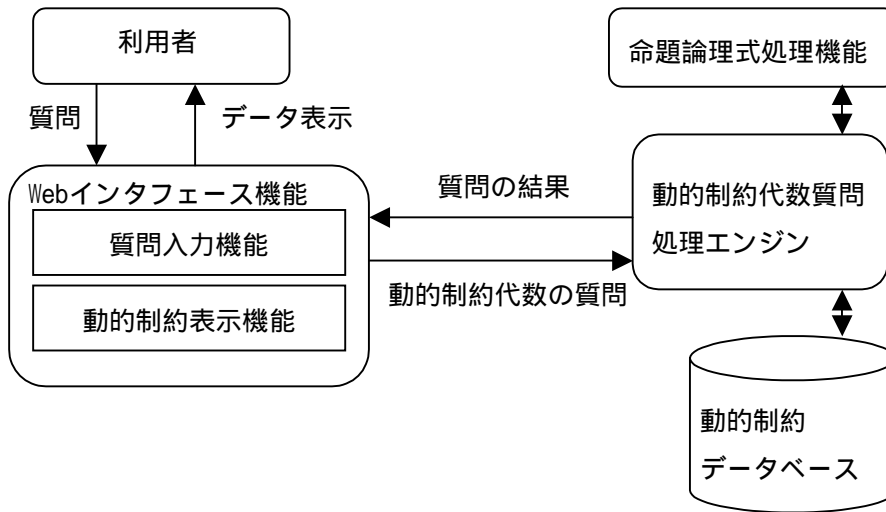


図4：システム全体図

このシステムの中で筆者は Web インタフェース機能の部分を担当し、実装を行った。Web インタフェースの実装には実装は Java を用いて行い、特にインタフェース部分には Java の GUI 用コンポーネントパッケージである Swing を用いた。

### 5.2. 画面一覧

#### i. Order

買い手のタブルの集合である制約関係を作成するための画面である。

#### ii. Query

Order で作成した買い手の制約関係をもとに質問を作成し、質問を発行する画面である。またその質問の結果を受け取ってタブルと動的制約を表示する画面でもある。



図6：Query 画面

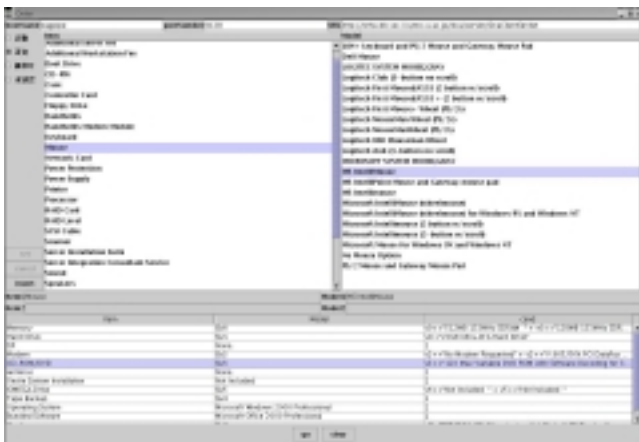


図5：Order 画面

### 5.3. シナリオ

PC 販売サイトでの想定される場面を挙げる。

#### I. カタログ作成

売り手が商品の一覧であるカタログを作成する。

#### II. カタログ選択

買い手がいくつかのカタログの中からカタログを選択する。

#### III. 利用者が希望する機能を選択

買い手が望む機能（例えば、品目、型）を選択する。

#### IV. 質問作成

買い手が質問を作成して、それを動的制約代数処理エンジンにわたす。

#### V. 制約表示

システムが質問の結果を買い手に対して表示する。

#### VI. カタログのタブル変更

買い手の質問やその結果に対して売り手がカタログのデータを変更する。

#### 5.4. 要求事項

このインタフェースの要求事項を場面ごとに挙げていく。

- ・ URL 入力機能

Web インタフェースなので URL 入力機能が必要。利用者は URL を入力する。

##### I. カタログ作成

- ・ カタログ作成支援機能

利用者は制約、品目、型、等の項目を入力または選択することによってタプルを作成する。もちろん直接入力することもできる。それらを集めてカタログを作成する。

##### II. カタログ選択

- ・ カタログ選択機能

利用者はいくつかのカタログの中から希望するカタログを選択する。

##### III. 利用者が希望する機能を選択

- ・ 買い手の制約関係作成機能

利用者は制約、品目、型を選択することによりタプルを作成する。それらを集めて制約関係を作成する。その制約関係において、正しい文法の制約式が得られることが必要である。また、利用者の選択した制約関係を確認、修正する機能も必要である。

##### IV. 質問作成

- ・ 質問作成支援機能

利用者が質問言語の文法を知らなくても質問を作成できるように質問の決まったパターンを示す。また質問言語の一覧を表示することで利用者が質問を作成するのを支援する。利用者が直接質問を入力することもできる。

##### V. 制約表示

- ・ 制約表示機能

質問によって得られた結果を利用者に対して表示する。

- ・ 検索機能

制約をわかりやすく表示する。例としては、利用者に関係のある制約だけを表示する。利用者の希望する情報を表示する。制約のあるパターンにまとめて表示する。

##### VI. カタログのタプル変更

- ・ タプル変更機能

利用者は変更したいタプルを選択し、その中のデータを変更する。

#### 5.5. 検証

著者は利用者が買い手に特化したインタフェースを作成した。よって上の ~ について検証する。

- ・ カタログ選択

今回は PC のみを扱うので、カタログ選択機能は必要が無く、初めから PC のカタログが選択してある状態になっている。

- ・ 利用者が希望する機能を選択

URL 入力機能

利用者は host 名、port 番号、URL を画面上部の入力部分にキーボードより直接入力する。特定の URL をデフォルトにしておくことにより、利用者がこれらを入力する作業を省略することも可能である。

制約関係作成支援機能

- ・ 買い手の制約を決める機能

画面左上に 4 つの制約パターンのボタンがある。次に選択する Item と Model の制約パターンを決めるため、利用者は制約を選択し、選択したボタンをクリックしなければならない。

- ・ 品目、型を決める機能

画面中央にある Item と Model のリストより、利用者の希望する Item と Model を選択する。選択した Item と Model はリストの下側にあるラベルに表示される。

- ・ タプルをまとめる機能

前の Item と Model の選択の後、利用者が選んだ Item と Model と制約をテーブルに挿入する。この作業を繰り返すことにより、テーブルがタプルの集合として表される。テーブルには文法の正しい制約式が挿入される。選択した制約関係はテーブルを見ることによって確認でき、適宜修正もできる。

- ・ 質問作成

質問作成支援機能

- ・ パターン質問入力

買い手の制約関係とカタログを結合する、制約が真となっているものを見つける、価格が 1 つに定まっている品目を見つける等の、決まった質問のパターンを質問入力部分に表示することができる。

- ・ コマンド質問入力

入力できるコマンドは何かがあるのかをプルダウンメニューより一覧にして利用者に表示する。利用者は使用するコマンドを選択し、その時引数もプルダウンメニューより選択できる。選択できない引数は直接入力することにより機能を補完する。

- ・ 直接入力

パターン入力やコマンド入力以外に、利用者は直接質問を質問入力部分に入力することができる。

- ・ 制約表示

- ・ タプル表示機能

制約だけの表示だと利用者が理解しにくいのでタプルを表示する。行を 1 つのタプル、列を属性とするテーブルを用いてタプルを表示する。

- ・ 制約表示機能

ダブルを選択したときにそれに対応した制約を表示する。制約式そのままの形式であると利用者が理解しにくいので、利用者に主な制約を文章の形に置換して表示している。

要求事項であげた買い手の機能は満足しているといえる。ただ、このインターフェースは必要最低限の機能しか保持していないので、利用者にとって利便性の高い機能の提供が必要である。例えば、制約表示のところで、制約のあるパターンにまとめて表示する（例えば、最安値の組合せを表示する）制約表示の高機能化などがあげられる。

## 5.6. 状態遷移

### I. Order 画面の遷移

- i. 初期状態 (図 7)
  - ii. 制約パターン選択
  - iii. Item, Model 選択 (図 8)
  - iv. テーブルに Item, Model 挿入 (図 9)
  - v. 最終状態 (図 10)
- ( ) Query 画面へ

以上のように遷移する。

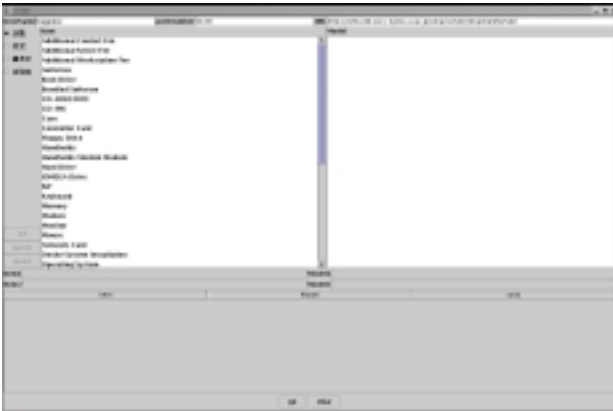


図 7：初期状態

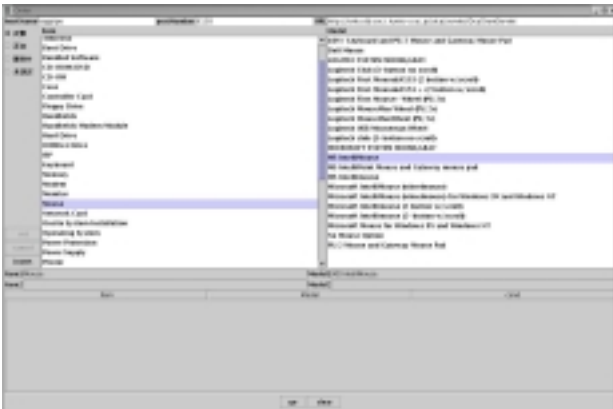


図 8：Item, Model 選択時

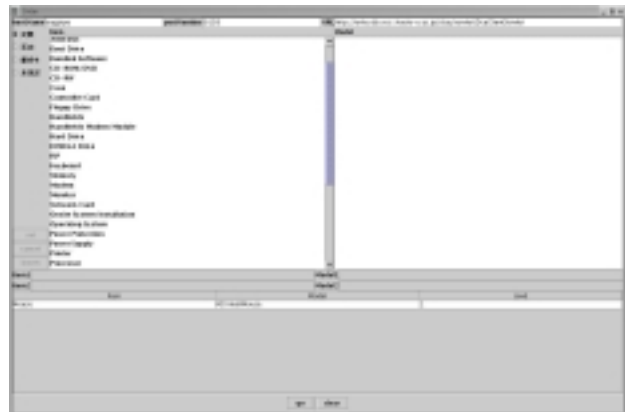


図 9：テーブル挿入時

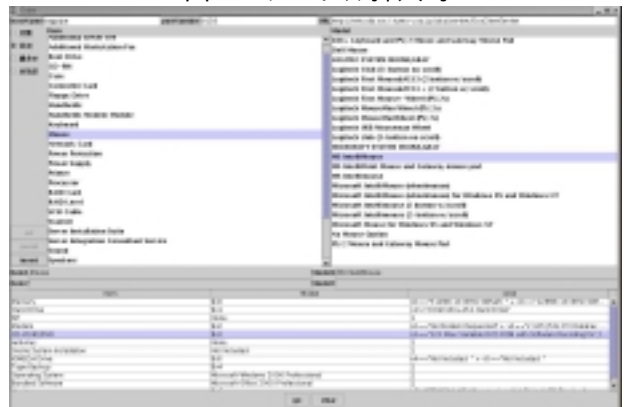


図 10：最終状態

### II. Query 画面の遷移

- i. 初期状態 (図 11)
  - ii. 質問入力状態 (図 12)
  - iii. 制約表示状態 (図 13)
- ( ) Order 画面へ

以上のように遷移する。

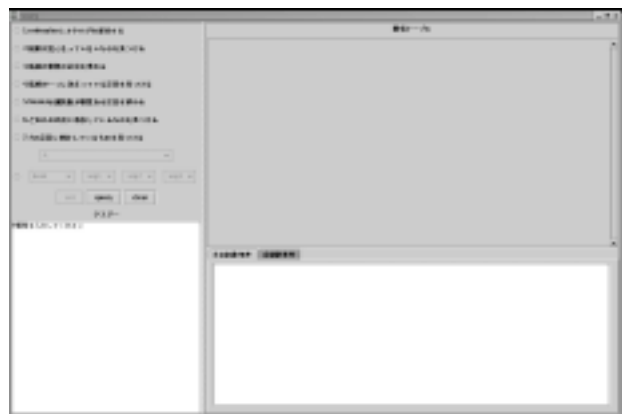


図 11：初期状態



図 1 2 : 質問入力状態



図 1 3 : 制約表示画面

## 第 6 章 終わりに

本論文ではまず動的制約についての制約関係を、例を挙げて述べた。そして制約組と動的制約代数、動的制約演算子をそれぞれ説明した。DCA は電子商取引サイト上で買い手と売り手の多種の複雑な制約の演算を行い、検索や検証を行うことができる。次に実際に存在する PC サイトから抽出した制約をいくつかの制約パターンに分類した。そしてベンチマーク質問を通じて DCA 質問処理システムの性能について評価を行った。また、この処理系を容易に扱うことのできる利用者インターフェースを作成した。

現在、このインターフェースは必要最低限の機能しか持っておらず利用者に高い利便性を提供しているとは言い難い。今後の課題は、この処理系の特徴をいかしたよりよいインターフェースを作ることである。例えば、質問によって返ってきた結果の中で、最も安い組合せを表示するとか、予算の中で買える組合せを表示するなど、機能拡充によりさらに使いやすいインターフェースを実現することが必要である。また、このインターフェースが利用者にとって使いやすい

いインターフェースかどうか被験者を用いて評価する必要がある。

## 参考文献

- [1]. M. Iwaihara, "Supporting Dynamic Constraints for Commerce Negotiations," *2nd Int. Workshop in Advanced Issues of E-Commerce and Web-Information Systems (WECWIS)*, IEEE Press, 12--20, June 2000.
- [2]. M. Iwaihara. "Matching and Deriving Dynamic Constraints for E-Commerce Negotiations," *Workshop on Technologies for E-Services*, (informal proceedings), Cairo, Sep. 2000.
- [3]. G. Kuper, L. Libkin and J. Paredaens, eds. "Constraint Databases," Springer Verlag, 2000.
- [4]. Juhnyoung Lee, Priscilla Wang and Ho Soo Lee, "A Visual One-Page Catalog Interface for Analytical Product Selection" *EC-Web 2001*.
- [5]. R.E. Bryant. "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Comput.*, Vol. C-35, No. 8, pp. 677-691, Aug. 1986.
- [6]. R. E. Bryant. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. *ACM Computing Surveys*, 24(3): 293--318, September 1992.
- [7]. E. Bertino, B. Catania and B. Chidlovskii. Indexing Constraint Databases by Using a Dual Representation. *Proc. Int. Conf. Data Engineering*, pp. 618--627, 1999.
- [8]. X. Cheng, G. Dong, T. Lau and J. Su. Data Integration by Describing Sources with Constraint Databases. *Proc. Int. Conf. Data Engineering*, pp. 374--381, 1999.
- [9]. V. Gaede, et al (eds.) *Constraint Databases and Applications*. (Proc. CDB'97 & Proc. CP'96 Workshops), Lecture Note in Computer Science, Springer, 1997.