

〔公 開〕

TR-M-0001

3次元仮想物体生成・編集システムにおける
ドメイン遷移メカニズムの実現

安 部 伸 治
Shinji ABE

1 9 9 6 2 . 8

A T R 知能映像通信研究所

3次元仮想物体生成・編集システムにおける ドメイン遷移メカニズムの実現

安部 伸治

要旨

著者らは、3次元形状に関する概念的な表現を体系化した知識を利用して、形状に関する言語的な表現をグラフィクスパラメータのレベルまで翻訳することによって、仮想物体の生成・編集を実現する手法の確立を目指している。このような手法では、形状に関する概念知識は多領域にわたるきわめて大規模なものが必要となるが、知識の構築や格納の効率を考慮すると領域(ドメイン)毎に個別化するほうが有利である。ところがドメイン個別化によって、それを利用する際に利用者が参照ドメインを意識的に選択しなければならないような状況では、利用者から見た表現は制約を受けることになりかねない。そこで本論文では、個別化された知識を統合的に利用して、見掛け上ひとつの多領域にわたる知識としての扱いを可能とするドメイン遷移メカニズムを提案する。本論文で実現した機能は次の通りである。(1) 仮想物体の生成・編集過程において、利用者が用いた説明や仮想物体の形状などから、文脈の変化を検出する機能。(2) 変化した文脈に適切に追従し、常に的確なドメインを参照する機能。

キーワード：オントロジー、仮想現実、文脈、知識の統合

1 まえがき

人間が行う創造的な活動、とりわけ芸術活動や商品企画などのような物理的な対象物が最終的な生産物となるような創造活動の比較的上流過程では、対象物に関する大雑把な完成イメージを構成する過程が存在すると考えられる。完成イメージは、絵画や立体的なモックアップなどで表現されることが多い。これらのようなモデルの構築過程では、頭の中に描かれた確固たるイメージが完成された後にモデル構築が開始されるというよりは、モデル構築の作業と同時進行的にイメージ創りが行われていると考えられる。たとえば、このようなプロセスの身近な例としては粘土細工などを挙げることができ、はじめは漠然としたメンタルイメージが粘土によって形状を構築する過程で徐々にはっきりしたものになり、メンタルイメージの完成と粘土によるモデルの完成がほぼ同時となるような状況を体験することができる。

さて、実際の商品企画などにおいては、企画を立案する立場の人間がメンタルイメージをモデルとしての的確に表現できる能力を持つことは必ずしも期待できないため、デザイナーなどのような造形に関する専門家を通じて可視化することが行われる。このような過程に計算

機環境を介在させ、企画立案を行う人間に対してモデルによる表現能力を与えることができれば、このプロセスの効率化にきわめて大きく貢献するだろう。

そこで著者らは、自然言語や手振りを用いたマルチモーダルなインタフェースによって、メンタルイメージをグラフィクスによる3次元仮想物体として可視化する手法の提案を行ってきた[1], [10], [11]。本研究は、人間どうしの対話によってメンタルイメージを伝達する際に用いている言語的な表現と、言語表現に付随して用いられる手振り表現を利用して、できるだけ自然に仮想物体を構築する手段を実現することを目指したものである。

本研究によって提案された手法[1], [10], [11]は、3次元形状に関する知識を2レベルオントロジーとして持ち、形状に関する言語的な概念表現をこれらの知識を利用して、最終的にはグラフィクスパラメータのレベルまで翻訳することによって、グラフィクスの生成・編集を実現する手法である。また、言語表現に対する補助的な表現として用いられる手振りについては、翻訳のプロセスにおいて、グラフィクスパラメータを直接制御する手段として利用している。

この手法では、メンタルイメージをモデルとして可視化するある意味での造形能力は、利用する知識内部の形状表現に依存するため、3次元形状に関して人間どうしの対話に現れる常識に相当する知識が必要であるという極論に容易に到達し得る。したがって、このような常識をひとつの知識として構築するよりは、むしろ知識をドメイン毎に個別化して、それぞれはドメインに閉じた知識のみを対象とした方が、知識の構築や格納効率などの面で有利である。

さて、ドメインを個別化することによって効率的に知識を構築・拡張することができても、それを利用する際に利用者が参照ドメインを意識的に選択しなければならないような状況では、利用者から見た表現はかえって制約を受けることになりかねない。そこで、個別化された知識を統合的に利用して、見掛け上ひとつの常識的な知識として利用できるようなメカニズムが必要となる。このようなメカニズムとして実現しなければならない機能は次の通りである。

1. 仮想物体の生成・編集過程において、利用者が用いた説明や仮想物体の形状などから、文脈の変化を検出する機能。
2. 変化した文脈に適切に追従し、常に的確なドメインを参照する機能。

文脈の検出については、自然言語処理などの分野では話題の同定などの問題として扱われ、主に概念的なレベルの相関を用いた手法(たとえば[9])が提案されている。しかしながら、本研究が対象とするような仮想物体の編集過程においては、言語による概念的な説明のみならず、グラフィクスレベルの操作(変形などの)によって文脈が変化する場合もあり、概念的なレベルのみによる文脈判定の方法が必ずしも適用できない。

そこで、本研究では仮想物体の生成・編集過程において、文脈が変化する状況を整理し、それぞれの状況に応じてレベルの異なる文脈判定を行うドメイン遷移メカニズムを提案する。提案する手法では特に、変形操作などのような概念レベルの文脈判定を用いることができない場合に対して、一般的な形状の記述レベルにおける文脈判定とドメイン遷移を実現する。

2 2レベルオントロジーを用いた3次元形状編集法の概要

人間どうしの対話において、ある形状に関するメンタルイメージを伝達しようとするとき、形状を表す一般的な表現(たとえば、“立方体”、“円筒”のように)を用いることはむしろ稀であり、形状をある特定の概念と結び付けて表現することが多い。たとえば、“だんじり”を思い描いている人が対話相手にこれを伝えようとするとき、つぎのような表現を用いたとする。

神輿のようなもので…担ぎ
棒がなくて…籠のところが
もっと高くて…それに、牛車
のような車輪がついてて… (例1)

このような例は、日常の会話の中では頻繁に現れるが、対象とする形状に関する直接的な表現が見いだせないために、対話相手と共有できると思われる幾つかのドメイン、ここでは“神輿”と“牛車”から形状をあらわす概念“神輿のようなもの”、“牛車の車輪”を例示して説明を行っている。また、この例の“だんじり”のように、対象ドメインに属する形状の概念が一般に知られていない場合には、この例のように別のドメインに属する概念を駆使してメンタルイメージを表現することが行われる。

このような説明に現れる、形状に関する概念はドメインに強く依存する。一方、“立方体”や“円柱”などのような直接的な形状の表現はドメインに依存しない一般的な形状表現である。そこで、本研究では形状に関する言語表現を翻訳するために用いる知識を2つのレベルに分割し、ドメイン依存の形状概念を概念レベルオントロジーとして体系化し、ドメイン非依存の形状記述を形状レベルオントロジーとして体系化する。

概念レベルオントロジーは、対象概念とそれを構成する部品の概念が木構造的に体系化されたものであり、構造上のリーフの位置には、もっともプリミティブな部品の概念が一般的な形状記述と対応されて記述されている。概念レベルオントロジーによって、形状に関する概念的な表現は一般的な形状の記述に翻訳される。

形状レベルオントロジーは、一般的な形状の記述をグラフィクスレベルに翻訳するための辞書として用いられる。本研究では、超2次関数 [8], [2] とそのパラメータを用いて、3次元形状をパラメトリックに記述する手法を採用しており、基本形状は8種類のパラメータからなる8次元空間上の領域として記述している。形状レベルオントロジーは、一般的な形状の表現をパラメータ空間に対応させるための辞書として機能する。また、超2次関数の各パラメータは、変形概念と対応させ易く、“もっと丸く”のような言語表現や、領域内の一点を“このくらい長く”などのような表現で手振りを用いて特定するなどの自然な編集操作を実現することができる。

図1に2レベルオントロジーを用いた形状概念の翻訳プロセスを示す。例題として、(例1)のような表現を翻訳してグラフィクスに反映させるプロセスを図1を用いて説明する。本説明では、ドメイン遷移はインプリメントされていないものとし対象ドメインを“神輿”に限定する。したがって、“牛車のような車輪”のような表現は翻訳の対象としない。

1. “神輿のような”の表現が入力されると、概念レベルオントロジーを利用して、神輿を構成する各部品の一般的な形状記述に翻訳する。
2. 各部品に関する一般的な形状記述を、形状レベルオントロジーを用いて超2次関数のパラメータの値の組に翻訳する。
3. 超2次関数にパラメータの値を代入して、各部品をグラフィクス表現に変換し、神輿を可視化する。
4. 続いて、“籠のところがもっと高く”の表現が入力されたとする。“高く”は、形状レベルオントロジーによってパラメータの値の更新と翻訳され、“籠”に相当する部品の超2次関数のパラメータを一部変更する。このプロセスでは、“このくらい高く”などのように、補助的な表現手段として手振りが用いられる場合があり、この場合には対象パラメータの値を手振りにしたがって直接制御する。
5. パラメータの変更をグラフィクスに反映させる。

以上のようなプロセスを繰り返すことにより、計算機環境と対話をしながらメンタルイメージを生成・編集することができる。また、本手法ではサイバークローブを用いてオブジェクトを直接操作するダイレクトマニピュレーションも実現されており、提示された部品の配置などに用いられる。

3 ドメイン遷移メカニズム

本章では、ドメイン遷移メカニズムの詳細について述べる。

ドメイン遷移メカニズムは、前章で述べた手法を利用して3次元仮想物体を編集している状況において、利用者が用いた概念表現による説明や仮想物体の形状の変化から文脈の変化を検出し、その時点で参照すべきもっとも適切なドメイン(の形状概念)を計算機が自動的に判断するメカニズムである。このメカニズムによって、利用者は現時点で参照すべきドメインや、その物理的な格納位置を意識する必要がなくなり、見掛け上は多くのドメイン横断的な形状概念としての扱いができるようになる。

前章でも述べたように、ドメインに依存するのは概念レベルの表現であり、形状レベルの表現は複数ドメインに共通して用いることができるため、遷移の対象となるのは概念レベルオントロジーである。

3.1 概念レベルオントロジーの構造

概念レベルオントロジーの構造の一部を、ドメイン“house”を例にとって図2に示す。図中において、“[concept]”は形状の概念を表し、“< shape >”は形状レベルへ引渡される一般的な形状記述である。構造“importances”に続く“(< shape > 0.48)”のような形状記述に付与された数値は、対応する形状記述が概念を表す典型性を表す数値であり、0.0 ~ 1.0のレンジを持つ。この値は、対象とする概念を表す形状として利用される頻度に応じて更新される。構造“remindings”に続く“([concept] 0.70)”のような概念に付与された数値は、“[concept]”という概念によって上位概念が連想される度合いを表す数値で、0.0 ~ 1.0のレンジを持つ。この値については、現在のところ複数のドメインに関して横断的に且つ客観的に値を与える、あるいは更新する手法を実現していないため、オントロジー構築の際に主観的な値を直接与えている。

3.2 文脈変化の検出と適切なドメインの判定

本節では、仮想物体生成・編集過程において文脈が変化する状況と、それぞれの状況に応じたドメイン遷移のメカニズムを構築する。

ドメイン遷移を必要とする状況には次のようなものが考えられる。

1. “牛車の車輪”のように現在参照しているドメインとは別のドメインが明示された場合。
2. 単に“車輪”と表現された場合のように、ドメインが明示されずにある概念表現が与えられ、且つ与えられた概念が現時点で参照しているドメインに存在しない場合。
3. 編集対象のオブジェクトに対して新たな部品が追加され、あるいは削除されたため、オブジェクト全体の対象ドメインが変わったとみなされる場合。

4. パラメトリックな表現レベルでオブジェクトの変形が行われ、対象オブジェクトが属するドメイン概念が別のドメインに変化したと判断される場合。

以下、それぞれの状況に応じたドメイン遷移メカニズムについて述べる。

3.2.1 ドメインが明示された場合の遷移

現在参照しているドメイン以外のドメイン名が明示的に与えられた場合には、明示的な文脈の変化と捉え、示されたドメインに遷移する。ただし、該当するドメインのオントロジーが存在しない場合には、ドメインが明示されなかった場合と同等に扱う。すなわち、オントロジーが存在する複数ドメインのなかで、与えられた概念が属するもっとも典型的と判断されるドメインに遷移させる(第3.2.2節参照)。

3.2.2 ドメインが明示されない概念による遷移

表現された概念が現在参照しているドメインに存在しない場合には、文脈が変化したものと判断し、次のような手続きによってドメインを遷移させる。

複数のドメインを次々と参照し、与えられた概念 “[concept]” をサーチする。“[concept]” が存在すれば、該当する概念の構造 “remindings” を参照し、“([domain-concept] value)” の値 “value” を調べる。次いで、もっとも大きな値 “value” を持つドメインに遷移させる。

このプロセスは、与えられた概念が属するもっとも典型的なドメインと判断されたドメインにおける概念を参照する過程を実現したものと捉えることができる。

一例として、“roof” という概念が与えられた場合について、図2のドメイン “[house]” 以外に、ドメイン “[car]” において次のような記述があるものとする。

```
[roof]---relations(part_of)-- > [car]
|
+-----exemplars-- > ([proto_roof], ...)
|
+-----importances-- > (({ingot} 0.48),
|                          ({cube} 0.35), ...)
|
+-----remindings-- > ([car] 0.50)
[proto_roof]---features-- > {ingot}
```

このとき、2つのドメイン (“[house]”, “[car]”) における “[roof]” の remindings の値 ([house] 0.70) と ([car] 0.50) を比較し、もっとも値の大きなドメイン “[house]” における “[roof]” を典型的な例として参照する。ただし、最大値を持つドメインが複数存在する場合には、利用者に選択を促すものとする。

3.2.3 部品の追加 / 削除によるドメイン遷移

第2章の(例1)において、ドメイン “神輿” は説明の出発点として単に例示に用いられただけであるから、続く説明 “担ぎ棒がなくて” といったように、ある概念を追加 / 削除した結果、全く別の文脈に変化している可能性があり、別のドメインに遷移させる必要が生じる場合がある。このようなケースでは、一部の部品の追加 / 削除によってその他すべての部品の概念が変化している場合があり、前述の2つのケースのような概念レベルの部品の記述 “[part_concept]” によるドメインの判定は困難である。したがってこのようなケースでは、形

形状レベルの記述“<part_shape>”を判定の材料とする。適切なドメインの判定は次のような定量的な基準を用いて行う。

現在対象としているオブジェクト“O”の部品に関して一般的な形状記述のレベルで見た集合を、

$$g_O = \{ \langle S_1^O \rangle, \langle S_2^O \rangle, \dots, \langle S_N^O \rangle \}$$

とする（“<S_i^O>”はオブジェクト“O”のi番目の部品の形状記述）。また、あるドメイン“D”に属する部品に関する形状と典型性をあらわす値の集合を、

$$G_D = \{ (\langle S_1^D \rangle V_{S_1}^D), (\langle S_2^D \rangle V_{S_2}^D), \dots, (\langle S_M^D \rangle V_{S_M}^D) \}$$

とする（“V_{S_i}^D”は、形状“<S_i>”がドメイン“D”の部品であるための典型性を表す数値）。このとき、オブジェクト“O”がドメイン“D”に属する確信度P(O ∈ D)を次の式(1)によって定義する。

$$P(O \in D) = \frac{1}{N} \cdot \sum_{i=1}^N f_{onD}(\langle S_i^O \rangle) \quad (1)$$

ただし、

$$f_{onD}(\langle S_i^O \rangle) = \begin{cases} V_{S_i}^D & \text{when } \langle S_i^O \rangle \in G_D \\ 0 & \text{when } \langle S_i^O \rangle \notin G_D \end{cases}$$

部品の追加/削除が行われたタイミングで、複数ドメインにおいて式(1)を用いて確信度を計算し、もっとも高いPの値を算出したドメインに対象ドメインを遷移させる。

例題として、図2における“家”の典型例“[proto_house]”にロッド（“<rectangular_rod>”）を加えた場合を取り上げて、確信度Pの変化を示す。

図2において、“[proto_house]”を形状記述のレベルまで辿ることにより、

$$g_{[proto_house]} = \{ \langle prism \rangle, \langle cube \rangle \}$$

であることがわかる。また、“[house]”のすべての部品を形状レベルまで辿ることにより、次の集合を得る。

$$G_{[house]} = \{ (\langle cylinder \rangle 0.48), (\langle cube \rangle 0.48), \\ (\langle cone \rangle 0.35), (\langle prism \rangle 0.48), \\ (\langle square_base_pyramid \rangle 0.40) \dots \}$$

この時点での、確信度P([proto_house] ∈ [house])を式(1)によって計算すると、次のようになる。

$$P([proto_house] \in [house]) = \frac{1}{2} \cdot (0.48 + 0.48) = 0.48$$

これにロッド“<rectangular_rod>”を追加したオブジェクトについて同様の計算を行うと、次のような結果を得る。

$$g_{[object]} = \{ \langle prism \rangle, \langle cube \rangle, \langle rectangular_rod \rangle \}$$

$$P([object] \in [house]) = \frac{1}{3} \cdot (0.48 + 0.48 + 0) = 0.32$$

この例では、ロッド“< rectangular_rod >”の追加によって、“[object]”がドメイン“[house]”に属する確信度が減少していることがわかる。

このように、部品の追加/削除が行われたタイミングで、以上のようなプロセスを他の複数のドメインにおいて実行し、もっとも大きな値の確信度を算出したドメインへ遷移させる。

3.2.4 部品の変形によるドメイン遷移

ここでは、パラメトリックな表現レベルでオブジェクトの変形が行われ、文脈が変化したために、対象オブジェクトが属するドメイン概念が別のドメインに変化する場合について述べる。

このようなケースでは、次のような判断に基づきドメインを遷移させる。

このケースでも前節と同様、形状レベルの記述“< part_shape >”を判定の材料とする。

第2章において述べたように一般的な形状の記述“< shape >”は、形状レベルオントロジーによって、超2次関数の8次元のパラメータ空間における領域に対応付けられている。これらのパラメータの更新によって、部品オブジェクトの形状記述“< part_shape >”が存在する領域が変化し、やがて別の形状記述“< new_part_shape >”の領域に入ったものとする。このケースは、対象オブジェクトの形状記述集合 $g_{[object]}$ から“< part_shape >”を削除し、さらに“< new_part_shape >”を追加した場合と同様に扱うことができ、前節の確信度 $P([object] \in [Domain])$ によるドメイン判定の手続きを経てドメインを遷移させる。

4 実験

第3章で述べたドメイン遷移メカニズムを CommonLisp 環境において実験システムとしてインプリメントした。本章では、実験システムを用いた幾つかの検証例を示す。

尚、第3.2.1節および第3.2.2節のケースの実験結果については自明と思われるので、本節では第3.2.3節と第3.2.4節のケースについてドメイン遷移実験の結果を示す。

4.1 実験システム概要

本節では、構築した実験システムについて主にインタフェースに焦点を当てて概説する。

図3は本実験システムのマンマシンインタフェースである。画面中央は作業空間で、利用者はこの部分を利用してメンタルイメージの構築を行う。利用者はサイバークロブを装着し、作業空間および部品提示空間(後述)におけるダイレクトマニピュレーション(オブジェクトを掴む、移動するなど)を実現できる。

画面左下は、人間が用いる概念レベルの言語表現に対するインタフェースである。(例1)に示したような“神輿”、“担ぎ棒”、“高く”、“牛車の車輪”などの表現はこのインタフェースを介して入力する。また、利用者に対するマシン側のリアクションとして、複数ドメインにおいて確信度 $P(object \in Domain)$ を計算した結果をこのインタフェースに提示する。尚、本実験のために4つのドメイン“[house](家)”, [mikoshi](神輿), [yatai](屋台), [joyosha](乗用車)”における概念レベルオントロジーを用意した。それぞれのドメインにおける確信度はテーブルとして一括して提示するようにした。

中央下は、超2次関数のパラメータをダイレクトに制御するインタフェースである。言語的な表現による変形以外に、このインタフェースを利用したパラメータ空間における連続的

な変形ができる。8つのバーは各々超2関数の8つのパラメータに対応し、指定した部品の形状の微調整などに利用できる。

画面右側は、部品提示空間である。システムは、確信度 P に基づき、現在参照しているドメイン以外に適切なドメインを見つけると、利用者に問い合わせたうえでドメイン遷移を実行する。遷移したドメインにおける部品とその概念を、部品提示空間に表示する。利用者はサイバークロップを用いたダイレクトマニピュレーションによって、部品を作業空間に取り込むことが可能で、この空間によりメンタルイメージの構築過程で利用者が必要とする部品(と概念)が常に供給される。

4.2 部品の追加によるドメイン遷移実験

現在対象としている“家”の概念に対して、“ロッド”を追加した場合について、対象ドメインを“[house]”から“[mikoshi]”に遷移させる実験の結果を示す。図3は、ロッドを追加する前の状態である。この状態での確信度はそれぞれのドメインに対して $P([object] \in [house]) = 0.60$, $P([object] \in [mikoshi]) = 0.34$, $P([object] \in [yatai]) = 0.34$, $P([object] \in [joyosha]) = 0.22$ である。

これに対してロッド“<rectangular_rod>”を追加した後の状態を図4に示す。この操作によって、確信度はそれぞれ $P([object] \in [house]) = 0.60$, $P([object] \in [mikoshi]) = 0.68$, $P([object] \in [yatai]) = 0.34$, $P([object] \in [joyosha]) = 0.21$ へと変化し、これによって対象ドメインが“[house]”から“[mikoshi]”に遷移した。このドメイン遷移にしたがって、右側に与えられる部品の構成や概念が変化していることがわかる。

4.3 部品の変形によるドメイン遷移実験

ここでは、対象オブジェクトがドメイン“[mikoshi]”に属している状態から、ロッドを徐々にディスク変形して屋台に遷移させる実験の結果を示す。

図5は変形操作の起点となる状態で、対象オブジェクトはドメイン“[mikoshi]”に属している。この状態での確信度はそれぞれ $P([object] \in [house]) = 0.3$, $P([object] \in [mikoshi]) = 1.00$, $P([object] \in [yatai]) = 0.30$, $P([object] \in [joyosha]) = 0.20$ である。

まず、変形対象部品を指定する。サイバークロップで部品に触れることによって、部品の指定が行える。また、神輿の部品概念である“担ぎ棒”という概念表現を利用でき、あるいは形状記述“<rectangular_rod>”をダイレクトに指定しても良い。超2次関数のパラメータを調整してロッドの断面を変形させ、次のような一般的な形状記述レベルでの遷移(パラメータ空間における領域の遷移)を実現する。

$$\text{"<rectangular_rod>"} \implies \text{"<cylindrical_rod>"}$$

この変形の直後の状態を図6に示す。この状態での確信度はそれぞれ $P([object] \in [house]) = 0.30$, $P([object] \in [mikoshi]) = 0.84$, $P([object] \in [yatai]) = 0.30$, $P([object] \in [joyosha]) = 0.21$ である。この時点では、それぞれの確信度の値に多少の変化はあるものの、最大値を持つドメインは変形の前と同じであるから、文脈の変化はなく、ドメイン遷移の必要はないものと判断される。

次に前述と同様の手続きを経てロッドの長さを短縮し、さらに断面のスケールアップを行った。この変形操作によって、次のような一般的な形状記述レベルでの領域遷移がおこなわれる。

$$\text{"<cylindrical_rod>"} \implies \text{"<disk>"}$$

この変形の直後の状態を図7に示す。この状態での確信度はそれぞれ $P([object] \in [house]) = 0.30$, $P([object] \in [mikoshi]) = 0.30$, $P([object] \in [yatai]) = 0.56$, $P([object] \in [joyosha]) = 0.40$ となり、対象ドメインは “[mikoshi]” から “[yatai]” に遷移した。

5 議論

本実験システムを一般の人を対象として展示し、デモンストレーションを行った。

本節では、デモにおける利用者からの幾つかの重要な指摘、特に今後の展開を考慮するうえで重要な指摘に基づいて考察を行う。

5.1 ドメイン遷移後の対象語彙の変化について

実験システムにおいて、現状ではドメインの遷移が起こると、対象オブジェクトを構成する部品の概念が変化し、それにしたがってある部品を指し示すために遷移前に用いていた概念表現が利用できなくなり、普段の人間どうしの対話に比較してきわめて不自然である、との指摘があった。直感的には、いちど利用されたドメインの概念レベルオントロジーを何らかの形で保存したうえで、複数ドメインにおける参照関係を維持する手法が考えれるが、これは極論すればドメイン個別に構築されたオントロジーを常識規模に再構築しなおすことと等価な作業であり、現実の問題として格納効率などの面でドメイン個別化の意義が失われる。むしろ、利用者による仮想物体構築の過程は、利用者独自の新たなドメイン “[my_domain]” の構築過程であると捉え、対象オブジェクトを構成する部品に対して利用者が用いた概念表現と部品形状記述を “[my_domain]” に次々と登録しておくほうが的確であるように思われる。

“[my_domain]” は、利用者が対象としているオブジェクトであるたったひとつの典型例のみによって構成されるため、次のような比較的単純な処理によって構築することができる。すなわち、利用者が対象としているオブジェクトを構成する部品形状とそれらに対して利用者が用いた概念表現を対応させて、次々と記録してゆけば良い。また、利用する際には、“[my_domain]” は遷移先のドメインオントロジーと常にペアで用いられるようにすれば良い。

5.2 空間的な配置関係について

現状の実験システムでは、複数部品の空間的な配置関係をオントロジーとして扱っていないために、個別の部品を提示するにとどまっている。対象ドメインにおけるドメイン概念の完成イメージ(部品を組上げた姿)は、現在のところグラフィクスレベルのデータとして登録したものを表示しており、変形操作の対象とならない。空間的な配置関係は形状レベルオントロジーに一般的な記述語彙とともに体系化するのが適当であると思われる。これは、超2次元関数の8次元空間を量子化したのと同様のアプローチで、3次元空間を量子化したうえで、それぞれの3次元領域に語彙を割り当てることによって構築できると考えられる。空間と記号表現との対応関係に関する研究としては[7]があり、3次元的な方向とそれに関する記号表現(たとえば“斜め前”、“左後ろ”など)の間の対応付けがおこなわれている。しかしながら、部品の配置関係は3次元的な方向のみで表現することは困難で、部品の姿勢が関係したり、また利用者の視点なども考慮しなければならない可能性などもあり、オントロジーに記述する客観的な表現を体系化するのはきわめて難しい。本手法においては、部品の配置はサ

イバークロブによるダイレクトマニピュレーションにより実現し、配置された結果はグラフィクスレベルのデータとして記録するようにしている。

6 関連研究

本研究によって提案されたドメイン遷移メカニズムは、複数のドメインにおける概念レベルオントロジーを対象としたある種の検索問題と捉えることができる。ただし、通常の情報検索とは異なり、利用者には遷移に対する意図は全く存在せず、遷移が必要な状況の判断や遷移先に関する判定がマシン側に委ねられているのが特徴である。

個別化されたオントロジーを一つの常識的な知識として利用するためには、むしろ情報検索的な処理を利用者に対して一切意識させないメカニズムとなるような工夫が必要である。そのためには、(例1)のような利用者の説明やオブジェクト編集過程における文脈の変化を検出し、変化した文脈に自動的に追従する必要がある。そこで本手法では、状況に応じて異なる3つの文脈判定を実現した。

このように本手法では、文脈の変化を遷移が必要な状況として判断し、遷移先のドメインの決定は、マシン側の処理として実現している。

文脈変化の検出以後の遷移先ドメインの決定法に関しては、幾つか関連の深い情報検索の手法があるので、以下それらについて述べる。

6.1 連想検索との関連について

第3.2.2節におけるドメイン遷移は、ある種の連想検索である。本手法では、概念レベルオントロジーの構造“reminders”のリーフの位置に記録された値をドメインを連想する重みとして用いた連想検索を実現している。したがって、現状では次のような問題が内在する。

- “reminders”の値の付与は、オントロジー構築過程のネックとなりかねないので、何らかの手法によって自動化する必要がある。

このような問題に対する直感的な解決法として、オントロジー構築の際には“reminders”の初期値(たとえば0.5)を与えておき、遷移先のドメインが決定された後にそのドメインにおける“reminders”の値を増加させる、などのようなある種の学習手法を採用することができる。ただし、

- ある概念から連想されるドメインは本来は個人や連想が行われる文脈に依存するものでありこれらに関する記述が必要となるが、個人や文脈の情報をオントロジーの中に記述すると、ドメインの個別性が失われかねない。

このようなことを考慮すると、連想のための重みを客観的な値として付与するより、利用者個別にドメイン横断的、且つ文脈に依存した連想知識をもつ方が良いかもしれない。

連想に関する知識を学習させる手法のひとつとして、アソシアトロン[4]が知られており、最近ではこの手法によりキーワードを連想的に拡張してテキストオブジェクトを検索する手法[5]などが提案されている。アソシアトロンは、複数キーワードの共起関係を学習させることができ、あるキーワード入力に対して、共起関係の高いキーワードを連想結果として出力する。この手法は学習のメカニズムも比較的単純で、且つ学習過程における人間の介入を一切必要としないため、本手法の連想メカニズムとして利用性が高い。

アソシアトロンを利用して、利用者が用いた概念と、的確な遷移と判断された場合のドメイン、さらに文脈としてその時点で提示されているすべての概念の共起関係を学習させれば、文脈に応じた的確な連想によるドメイン遷移を実現できる可能性がある。

6.2 3次元形状の類似検索手法との関連について

第3.2.3節および第3.2.4節のドメイン遷移メカニズムは、一般的な形状記述のレベルにおける類似検索の手法と捉えることができる。

3次元形状に関する類似検索を実現した研究例として、[2]が挙げられる。[2]では、本手法と同じく超2次元関数を形状の記述に用いており、超2次元関数のパラメータ空間における距離の概念を導入して、直接的な形状類似検索を実現している。超2次元関数は、様々な形状を8次元の連続的な変量空間に写像させることができるため検索の精度もきわめて高く、且つ8種類のパラメータと変形の概念が直接的に対応付け易いため、検索キーとなるオブジェクト生成が容易に行えることが示されている[2]。

しかしながら[2]の手法は基本的に、単一形状どうしの比較を目的としたもので、本手法における比較のように、複数部品の形状の組み合わせに関して比較を行う目的に利用するのは不利である。そこで本手法では、超2次元関数のパラメータ空間を量子化した上で、形状レベルオントロジーを用いて一般的な形状に関する語彙レベルの表現“< shape >”に対応させ、複数部品に対するこのような記述の集合(g_0)による間接的な類似検索を実現している。

7 まとめと今後の展開

本論文では、3次元形状に関してドメイン毎に個別化して構築した概念レベルオントロジーを、統合的に利用してひとつの常識的なオントロジーとしての扱いが可能となる、ドメイン遷移メカニズムを提案した。また、本手法をCommonLisp環境においてインプリメントし、デモンストレーションを行った。さらに、利用者からの指摘に基づいて現時点での問題点を明らかにした。以下、それらの指摘に基づいて今後に残された課題について述べる。

実験に関する考察や議論の節で述べたように、今後の検討課題の中心は、概念レベルオントロジーの効率的な構築手法の確立である。特に、実験に関する考察の最初の指摘は重要で、新たなドメインのオントロジー(第5節においては“[my_domain]”)の構築のために、すでに構築されたオントロジーとドメイン遷移メカニズムを効果的に利用した、ドメインオントロジー構築ツールが必要である。

人間どうしの対話には、説明しようとするメンタルイメージの属するドメインが対話相手には理解できない状況において、共有できる様々なドメインにおいてメンタルイメージと類似する形状をもつ概念を駆使して説明する過程が存在する。このような過程をドメイン遷移を用いて実現し、最終的に出来上がったオブジェクトとその部品に対して概念を付与するような、インタラクティブなオントロジー構築ツールを実現すれば、取りあえず初期の目的は達成される。ただし、この時点で構築されるオントロジーは利用者が造り上げた、たったひとつの典型例のみからなるため、ドメインの表現力はきわめて未熟である。これを様々な事例(後に構築され当該ドメインに属すると判断されたオブジェクトや、他の知識から抽出した概念構造が対象となる)をもとに、オントロジーを効率的に補強するメカニズムを実現しなければならない。

そのための具体的なアプローチや要素技術としては、知識ベースとデータベースの統合利用(たとえば[3])など参考になる研究があり、今後最重要課題として取り組む必要があると考える。

参考文献

- [1] 安部伸治, 吉田美寸夫, ティヘリノ ジュリ, 岸野文郎, “マルチモーダル3-D形状編集シ

ステムの実現”, 信学技報 HC94-57, pp.39-46, 1994.

- [2] Horikoshi, T. and Kasahara, H., “3-D Shape indexing language”, in Proc. of the 1990 International Conference on Computers and Communications, pp.493-499, 1990.
- [3] 伊藤 秀昭, 福村 晃夫, “知識ベース-データベース統合化ツール IKD の構造とその利用”, 人工知能学会誌, Vol.8, No.1, pp.102-113, 1993.
- [4] 中野 馨, “アソシアトロン - 連想記憶のモデルと知的情報処理 -”, 昭晃堂, 1979.
- [5] 西本一志, 安部伸治, 宮里勉, 岸野文郎, “連想記憶を用いた発散的思考支援システムにおける提供情報の分野制御の試み”, 計測自動制御学会 第15回システム工学部会研究会「発想支援技術」(於 東京工業大学百年記念館), pp.17-24, 1994.
- [6] Lenat, D.B. and Guha, R.V., “Cyc: Toward programs with common sense”, Communications of the ACM, vol.33, No.8, pp.30-49, 1990.
- [7] 望月 研二, 岸野 文郎, “言語による3次元モデル世界におけるシーン検索・記述”, 1991年度人工知能学会全国大会(第5回)予稿集, pp.629-632, 1991.
- [8] Pentland, A.P., “Perceptual organization and the representation of form”, Artificial Intelligence, vol.28, pp.293-331, 1986.
- [9] 竹下 敦, “対話のインタラクション構造を用いた話題の認識”, 情報処理学会研究技術報告 87-10, pp.75-82, 1992.
- [10] Tijerino, Y.A., Abe, S., Miyasato, T. and Kishino, F., “What You Say Is What You See, -Interactive generation, manipulation and modification of 3-D shapes based on verbal descriptions-”, Artificial Intelligence Review Journal, Vol.8, No.2, pp.123-142, 1994.
- [11] Tijerino, Y.A., Abe, S. and Kishino, F., “Intuitive graphic representation of mental images in a virtual environment through natural language”, to be published in Proc. of AAAI Fall Symposium Series, Cambridge, Mass., November 1995.

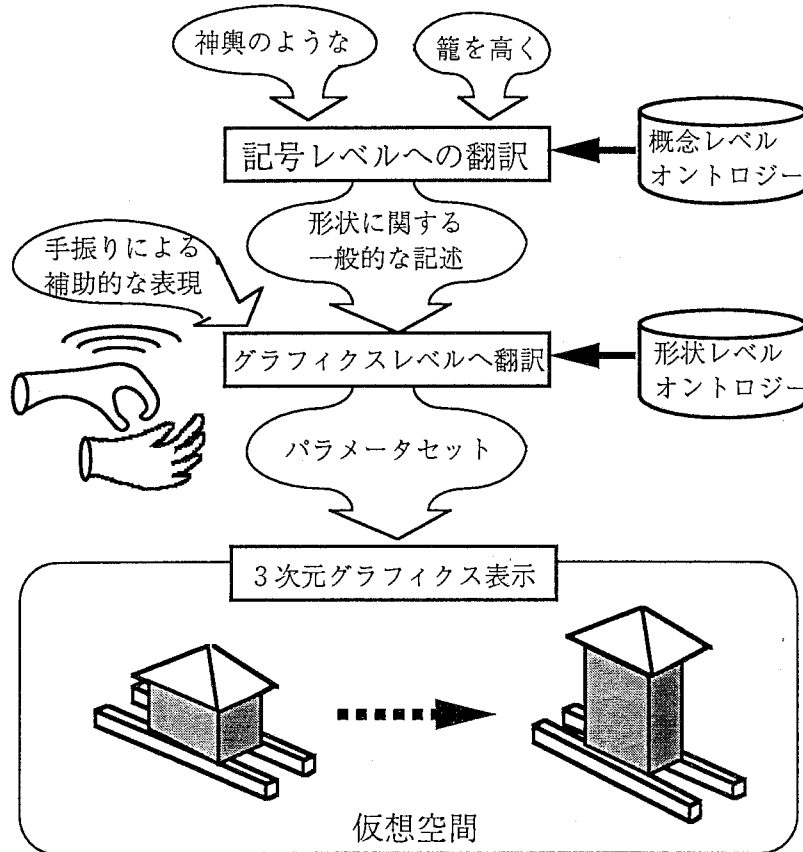


図 1: 2 レベルオントロジーを用いた形状概念の翻訳プロセス
 Translation process of concept level explanation about 3-D shape

```

[house]--relations(has_parts)-->+---[walls]
|
|           |
|           +---[roof]
|
+-----exemplars-->([proto_house], [humble_house], [dome_house], [indian_house],
[walls]--relations(part_of)-->[house]
|
+-----exemplars-->([proto_walls], [walls1], [walls2], ...)
|
+-----importances-->((< cylinder > 0.48), (< cube > 0.48), ...)
|
+-----reminders-->([house] 0.67)
[roof]---relations(part_of)-->[house]
|
+-----exemplars-->([proto_roof], [roof1], [roof2], ...)
|
+-----importances-->((< cone > 0.35), (< prism > 0.48), (<
square_base_pyramid > 0.40), ...)
|
+-----reminders-->([house] 0.70)
[proto_house]--features-->+---[proto_roof]
|
|           |
|           +---[proto_walls]
|
+-----typicality-->([house] 0.85)
[proto_roof]---features--> < prism >
[proto_walls]--features--> < cube >

```

ここで, [concept] : 形状の概念.
 < shape > : 形状レベルへ引渡される一般的な形状記述.

図 2: 概念レベルオントロジーの構造 (部分)

Concept level ontology

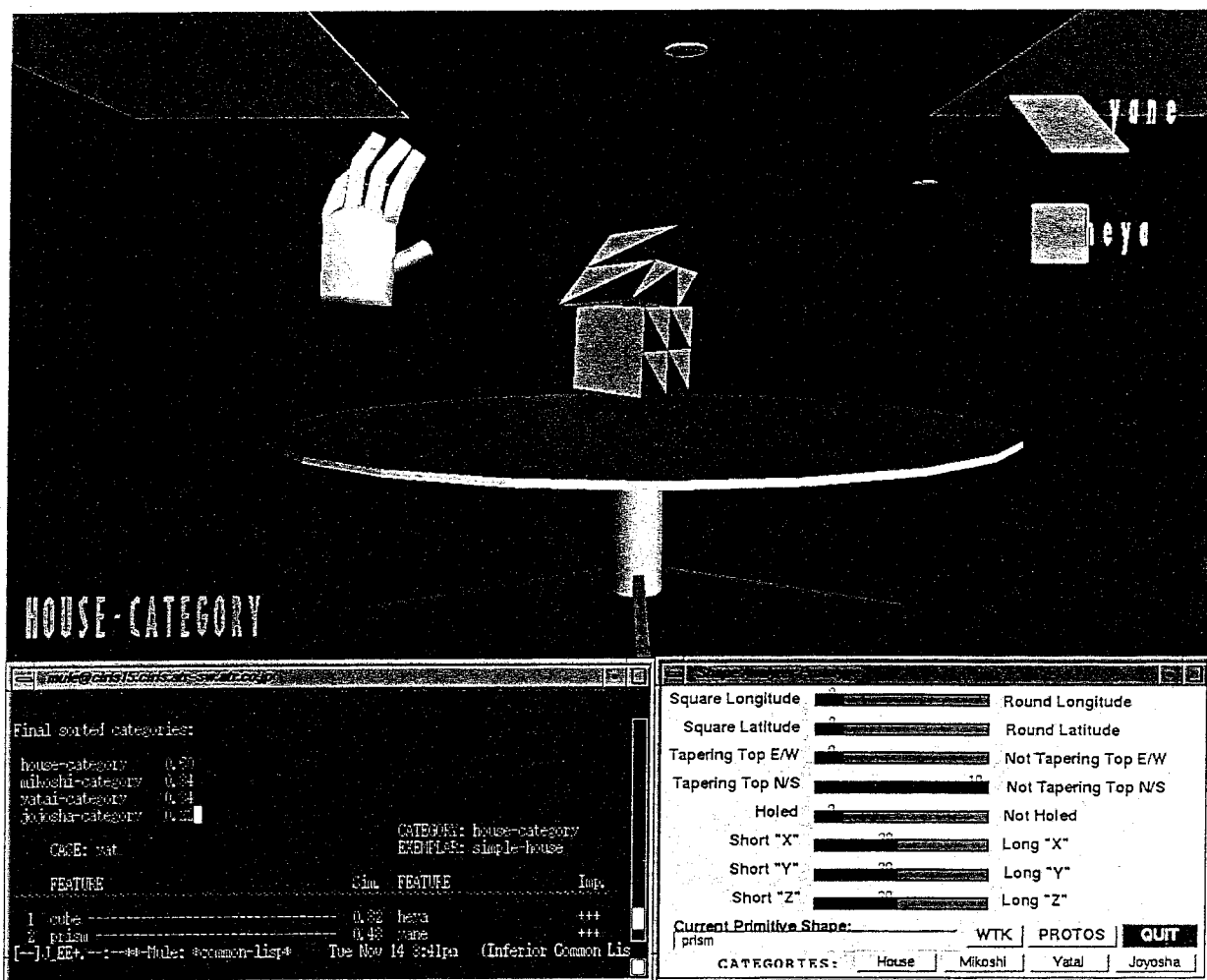


図 3: 対象オブジェクトの概念が “[house]” であると判断されている状態
 Situation where the 3-D object belongs to the domain “[house]”

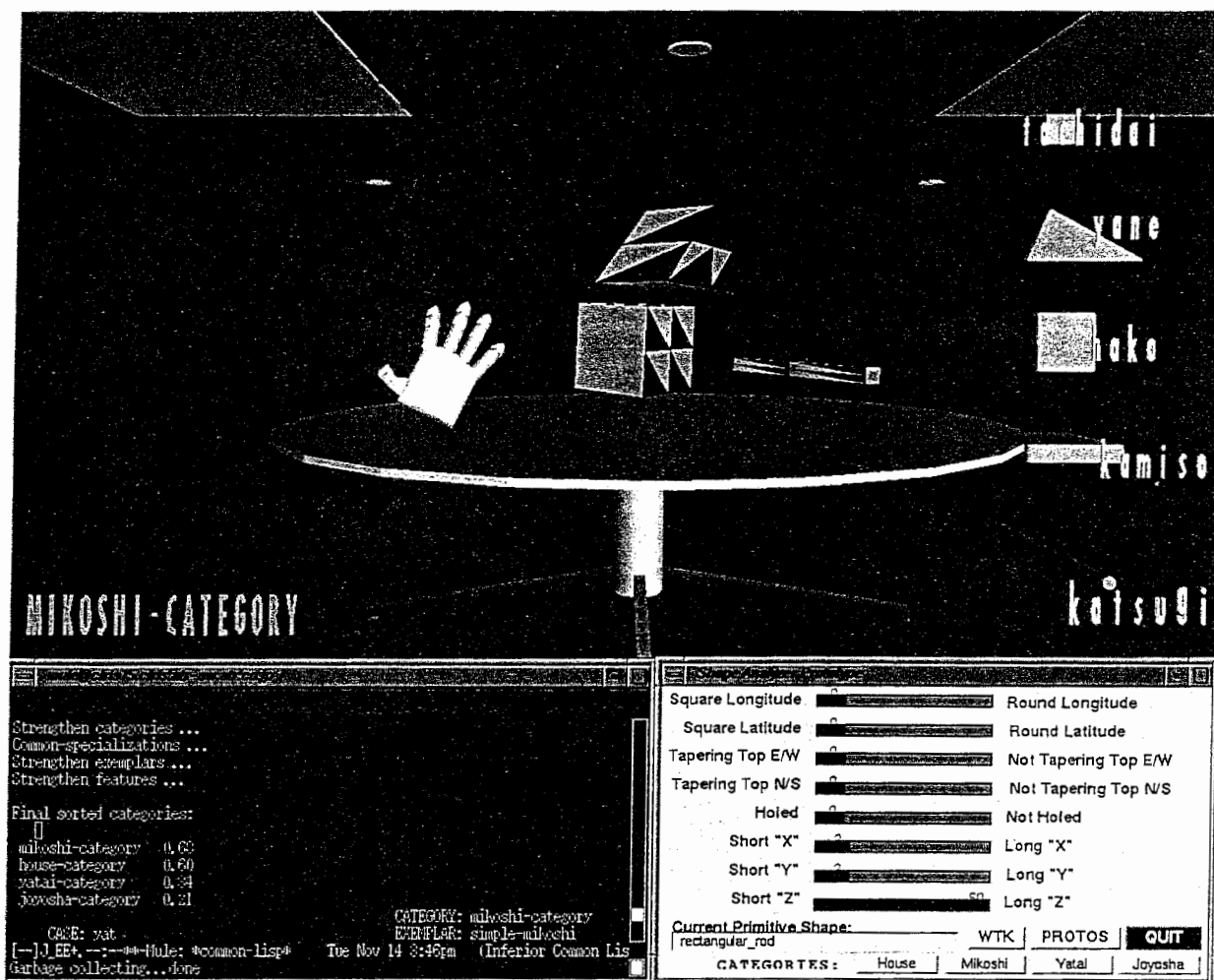


図4: ロッド“< rectangular_rod >”の追加によって、対象ドメインが“[mikoshi]”に遷移した直後の状態

Situation after the domain transition into “[mikoshi]” by addition of “< rectangular_rod >”

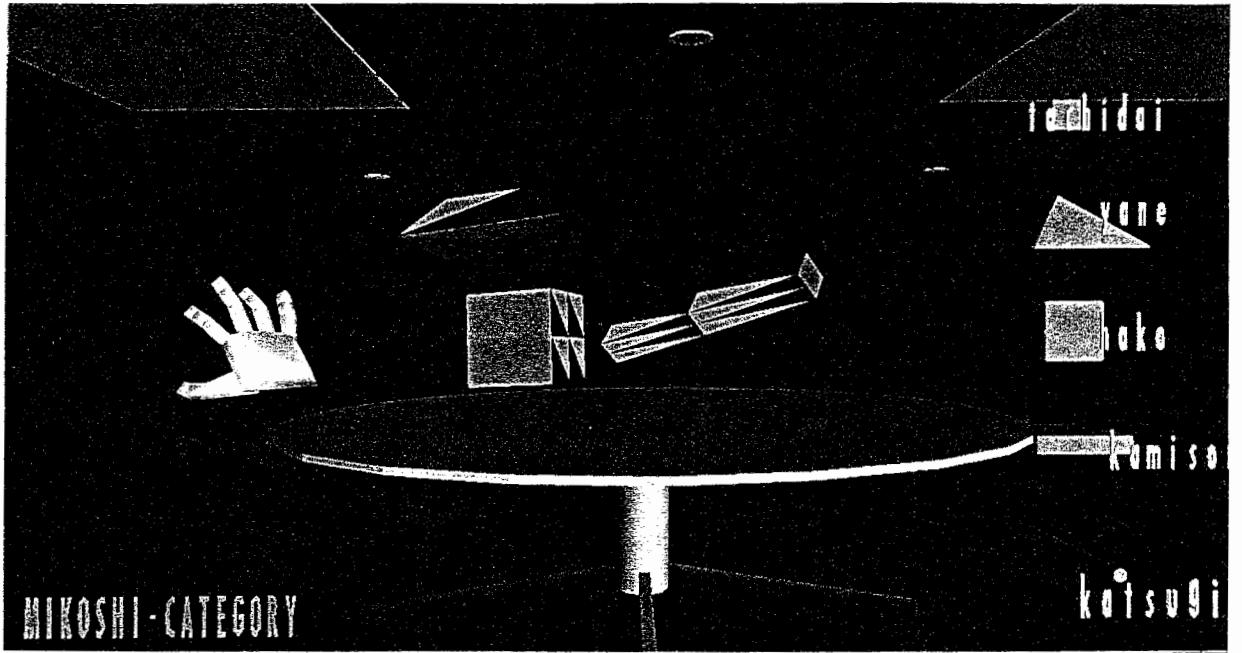


図 5: 変形操作の起点となる状態: 対象オブジェクトはドメイン “[mikoshi]” に属している。
Situation where the 3-D object belongs to the domain “[mikoshi]”

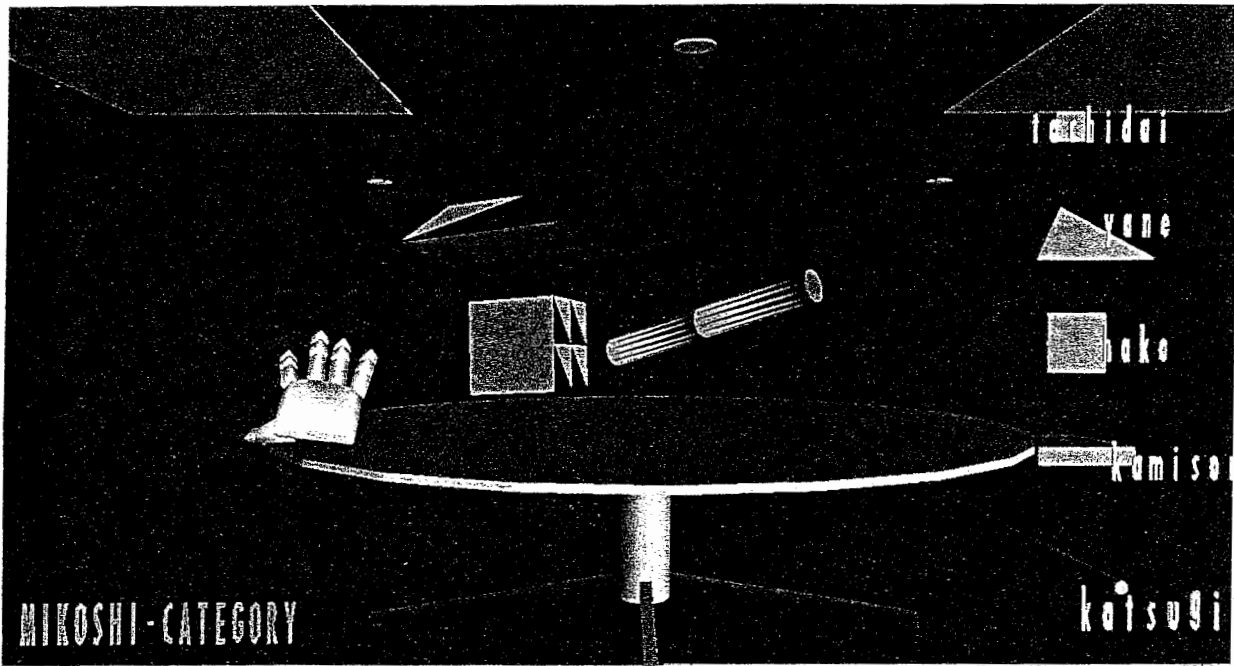


図 6: ロッド断面の変形によって“< rectangular_rod >”から“< cylindrical_rod >”に変化させた直後の状態

Situation where “< rectangular_rod >” has been deformed into “< cylindrical_rod >”

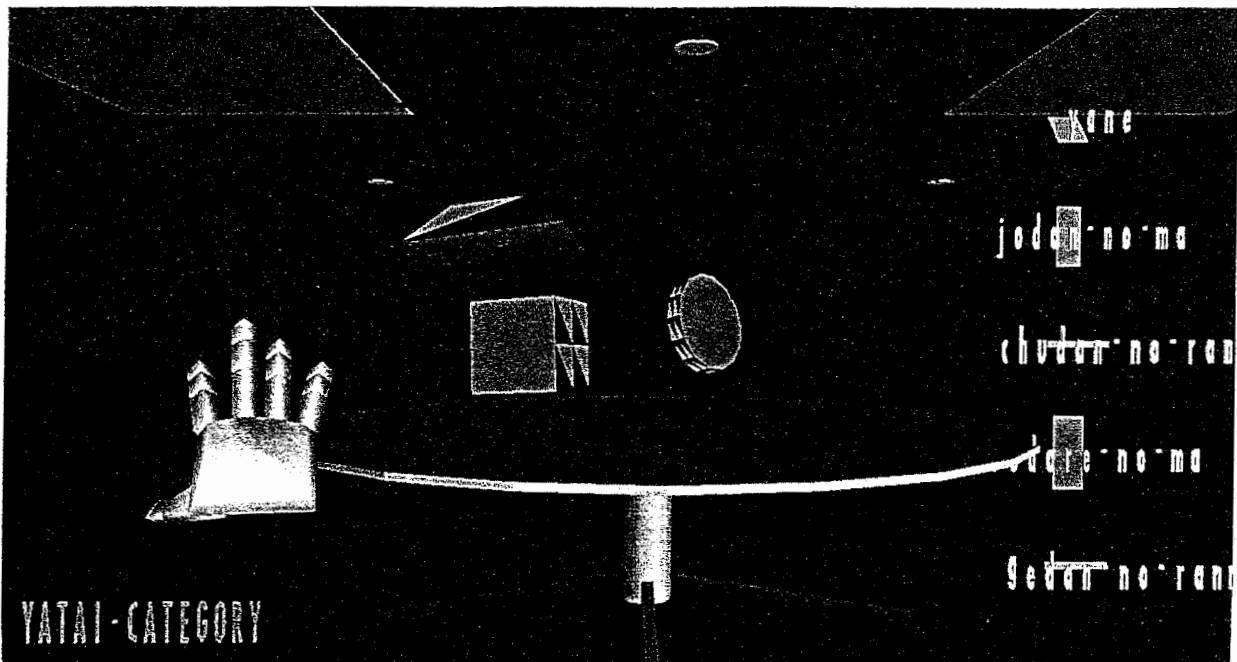


図 7: ロッドの長さの変更によって“< cylindrical_rod >”から“< disk >”に変化させた直後の状態

Situation where “< cylindrical_rod >” has been deformed into “< disk >”

Appendix (概念レベルオントロジー)

```

:domain "Mikoshi"
;

(term :name      mikoshi
      :importances ((rectangular_rod 0.48) (cube 0.48) (pyramid 0.34) (rectangular_sheet 0.48)
                    (cylindrical_rod 0.48) (prism 0.48))
      :exemplars  (simple-mikoshi another-simple-mikoshi)
      :relations  (((mikoshi always has part katsugibo) "katsugibo is always a part of mikoshi")
                    ((mikoshi usually has part kamisori) "kamisori is usually a part of mikoshi")
                    ((mikoshi always has part kago) "kago is always a part of a mikoshi")
                    ((mikoshi always has part yane) "Yane is always a part of a mikoshi")
                    ((mikoshi always has part tachidai) "tachidai is always a part of a mikoshi")))

(term :name      tachidai
      :reminders  (mikoshi 0.70)
      :importances ((cube 0.48))
      :exemplars  (simple-tachidai)
      :relations  (((tachidai always is part of mikoshi) "tachidai is always a part of a mikoshi")
                    ((tachidai sometimes is inferred from cube) "cube sometimes suggests tachidai")))

(term :name      yane
      :reminders  (mikoshi 0.68)
      :importances ((prism 0.48) (pyramid 0.48))
      :exemplars  (simple-mikoshi-yane)
      :relations  (((yane always is part of mikoshi) "Yane is always a part of a mikoshi")
                    ((yane sometimes is inferred from pyramid) "pyramid sometimes suggests yane")
                    ((yane sometimes is inferred from prism) "prism sometimes suggests yane")))

(term :name      kago
      :reminders  (mikoshi 0.70)
      :importances ((cube 0.48))
      :exemplars  (simple-kago)
      :relations  (((kago always is part of mikoshi) "kago is always a part of a mikoshi")
                    ((kago sometimes is inferred from cube) "cube sometimes suggests kago also")))

(term :name      kamisori
      :reminders  (mikoshi 0.56)
      :importances ((rectangular_sheet 0.48))
      :exemplars  (simple-kamisori)
      :relations  (((kamisori usually is part of mikoshi) "kamisori is usually a part of mikoshi")
                    ((kamisori sometimes is inferred from rectangular_sheet) "rectangular_sheet sometimes suggests kamisori")))

(term :name      katsugibo
      :reminders  (mikoshi 0.54)
      :importances ((cylindrical_rod 0.48) (rectangular_rod 0.48))
      :exemplars  (left-katsugibo right-katsugibo round-katsugibo)
      :relations  (((katsugibo always is part of mikoshi) "katsugibo is always a part of mikoshi")
                    ((katsugibo sometimes is inferred from rectangular_rod) "rectangular rod sometimes suggests katsugibo")
                    ((katsugibo sometimes is inferred from cylindrical_rod) "MIV-USER" "cylindrical rod usually suggests katsugibo")))

(term :name      simple-mikoshi
      :comment    "(the exemplar of mikoshi)"
      :category   mikoshi
      :features   (tachidai yane kago kamisori katsugibo)
      :typicality 1.75)

(term :name      simple-tachidai
      :comment    "(an exemplar of a simple tachidai)"
      :category   tachidai
      :features   (cube)
      :typicality 1.00)

(term :name      simple-mikoshi-yane
      :comment    "(an exemplar of a simple mikoshi yane)"
      :category   yane
      :features   (pyramid)
      :typicality 1.00)

(term :name      simple-kago
      :comment    "(an exemplar of a simple mikoshi kago)"
      :category   kago
      :features   (cube)
      :typicality 1.00)

(term :name      simple-kamisori
      :comment    "(an exemplar of simple mikoshi kamisori)"

```

```

:category    kamisori
:features    (rectangular_sheet)
:typicality 1.00

(term :name    left-katsugibo
:comment     "(an exemplar of a left katsugibo for a simple mikoshi)"
:category     katsugibo
:features     (rectangular_rod)
:typicality  1.00)

(term :name    right-katsugibo
:comment     "(an exemplar of a right katsugibo for a simple mikoshi)"
:category     katsugibo
:features     (rectangular_rod)
:typicality  1.00)

(term :name    another-simple-mikoshi
:comment     "(another simple mikoshi exemplar)"
:category     mikoshi
:features     (pyramid cube rectangular_rod)
:typicality  1.00)

(term :name    round-katsugibo
:category     katsugibo
:features     (cylindrical_rod)
:typicality  1.00)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;          Domain "Yatai"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(term :name    yatai
:importances ((prism 0.48) (rectangular_sheet 0.48) (ingot 0.48) (disk 0.77) (ingot 0.48))
:exemplars   (simple-yatai danjiri danjiri-simple-variation)
:relations   (((yatai always has part sharin) "sharin is always a part of yatai")
              ((yatai sometimes has part gedan-no-ma) "gedan-no-ma is sometime a part of a yatai-catagory")
              ((yatai usually has part sudare-no-ma) "sudare-no-ma is usually a part of the yatai")
              ((yatai sometimes has part gedan-no-ranma) "gedan-no-ranma is sometimes po of the yatai category")
              ((yatai usually has part chudan-no-ranma) "chudan-no-ranma is usually a part of the yatai
              (chudan-no-ranma usually part of yatai)")
              ((yatai always has part jodan-no-yane) "jodan-no-yane is always a part of a yatai")
              ((yatai always has part jodan-no-ma) "jodan-no-ma is always a part of yatai"))))

(term :name    jodan-no-yane
:reminders   (yatai 0.70)
:importances ((prism 0.48))
:exemplars   (simple-jodan-no-yane)
:relations   (((jodan-no-yane always is part of yatai) "jodan-no-yane is always a part of a yatai")
              ((jodan-no-yane sometimes is inferred from prism) "prism sometimes suggests jodan-no-yane"))))

(term :name    chudan-no-ranma
:reminders   (yatai 0.56)
:importances ((rectangular_sheet 0.48))
:exemplars   (simple-chudan-no-ranma)
:relations   (((chudan-no-ranma sometimes is inferred from rectangular_sheet) "a rectangular_sheet sometimes suggests chudan-no-ranma")
              ((chudan-no-ranma usually is part of yatai) "chudan-no-ranma is usually a part of the yatai
              (chudan-no-ranma us po yatai)"))))

(term :name    gedan-no-ranma
:reminders   (yatai 0.42)
:importances ((rectangular_sheet 0.48))
:exemplars   (simple-gedan-no-ranma)
:relations   (((gedan-no-ranma sometimes is inferred from rectangular_sheet) "rectangular_sheet occasionally suggests gedan-no-ranma")
              ((gedan-no-ranma sometimes is part of yatai) "gedan-no-ranma is sometimes po of the yatai category"))))

(term :name    sudare-no-ma
:reminders   (yatai 0.56)
:importances ((ingot 0.48))
:exemplars   (simple-sudare-no-ma)
:relations   (((sudare-no-ma strongly implies chudan-no-ma) "sudare-no-ma strongly implies chudan-no-ma")
              ((sudare-no-ma usually is part of yatai) "sudare-no-ma is usually a part of the yatai")
              ((sudare-no-ma sometimes is inferred from ingot) "ingot sometimes suggests sudare-no-ma"))))

(term :name    gedan-no-ma
:reminders   (yatai 0.42)
:importances ((cube 0.48))

```

```

:exemplars (simple-gedan-no-ma)
:relations (((gedan-no-ma sometimes is inferred from cube) "cube occasionally suggests gedan-no-ma")
            ((gedan-no-ma sometimes is part of yatai) "gedan-no-ma is sometime a part of a yatai"))

(term :name      sharin
     :reminders (yatai 0.68)
     :importances ((disk 0.77))
     :exemplars (simple-yatai-sharin)
     :relations (((sharin always is inferred from disk) "a disk always suggests sharin")
                 ((sharin always is part of yatai) "sharin is always a part of yatai")))

(term :name      simple-yatai
     :comment     "(an exemplar of a yatai category)"
     :category    yatai
     :features    (jodan-no-yane chudan-no-ranma gedan-no-ranma sudare-no-ma
                  gedan-no-ma sharin)
     :typicality 2.75)

(term :name      jodan-no-ma
     :reminders (yatai 0.70)
     :importances ((ingot 0.48))
     :exemplars (simple-jodan-no-ma)
     :relations (((jodan-no-ma always is part of yatai) "jodan-no-ma is always a part of yatai")
                 ((jodan-no-ma sometimes is inferred from ingot) "ingot occasionally suggests jodan-no-ma"))))

(term :name      danjiri
     :category    yatai
     :features    (jodan-no-yane chudan-no-ranma gedan-no-ranma sudare-no-ma
                  gedan-no-ma sharin jodan-no-ma)
     :typicality 1.00)

(term :name      chudan-no-ma
     :relations  (((chudan-no-ma sometimes is implied by sudare-no-ma) "sudare-no-ma strongly implies chudan-no-ma"))))

(term :name      simple-jodan-no-yane
     :category    jodan-no-yane
     :features    (prism)
     :typicality 1.00)

(term :name      simple-jodan-no-ma
     :comment     "(an exemplar of a simple jodan-no-ma)"
     :category    jodan-no-ma
     :features    (ingot)
     :typicality 1.00)

(term :name      simple-chudan-no-ranma
     :comment     "(an exemplar of a simple chudan-no-ranma)"
     :category    chudan-no-ranma
     :features    (rectangular_sheet)
     :typicality 1.00)

(term :name      simple-sudare-no-ma
     :comment     "(an exemplar of a simple sudare-no-ma)"
     :category    sudare-no-ma
     :features    (ingot)
     :typicality 1.00)

(term :name      simple-gedan-no-ranma
     :comment     "(an exemplar of a simple gedan-no-ranma)"
     :category    gedan-no-ranma
     :features    (rectangular_sheet)
     :typicality 1.00)

(term :name      simple-gedan-no-ma
     :comment     "(an exemplar of simple gedan-no-ma)"
     :category    gedan-no-ma
     :features    (cube)
     :typicality 1.00)

(term :name      simple-yatai-sharin
     :comment     "(an exemplar of a simple yatai-sharin)"
     :category    sharin
     :features    (disk)
     :typicality 1.00)

(term :name      danjiri-simple-variation

```

```

:category yatai
:features (prism rectangular_sheet cube disk)
:typicality 1.00)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Domain "Joyosha"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(term :name joyosha
:importances ((ingot 0.48) (cube 0.48) (disk 0.42))
:exemplars (sedan my-car)
:relations (((joyosha sometimes has part tyre) "tyre is sometimes part of a joyosha")
((joyosha always has part chassis) "chassis is always a part of a joyosha")
((joyosha usually has part yane) "yane is usually a part of a joyosha")))

(term :name chassis
:reminders (joyosha 0.70)
:importances ((cube 0.48) (ingot 0.48))
:exemplars (simple-jidosha-chassis)
:relations (((chassis always is part of joyosha) "chassis is always a part of a joyosha")
((chassis sometimes is inferred from ingot) "ingot sometimes suggests chassis")
((chassis sometimes is inferred from cube) "cube sometimes suggests chassis")))

(term :name tyre
:reminders (joyosha 0.70)
:importances ((disk 0.48))
:exemplars (simple-tyre)
:relations (((tyre always is part of joyosha) "tyre is always a part of a joyosha")
((tyre sometimes is inferred from disk) "disk sometimes suggests tyre")))

(term :name yane
:reminders ((joyosha 0.56))
:importances ((ingot 0.48) (cube 0.48))
:exemplars (jidosha-yane)
:relations (((yane usually is part of joyosha) "yane is usually a part of a joyosha")
((yane sometimes is inferred from ingot) "ingot sometimes suggests yane")
((yane sometimes is inferred from cube) "cube sometimes suggests yane")))

(term :name sedan
:comment "(an exemplar a joyosha)"
:category joyosha
:features (yane chassis tyre)
:typicality 2.00)

(term :name jidosha-yane
:comment "(an exemplar of a jidosha-yane)"
:category yane
:features (ingot)
:typicality 1.25)

(term :name simple-jidosha-chassis
:comment "(an exemplar of a simple jidosha ingot)"
:category chassis
:features (ingot)
:typicality 1.00)

(term :name simple-tyre
:comment "(an exemplar of a tyre)"
:category tyre
:features (disk)
:typicality 1.00)

(term :name my-car
:category joyosha
:features (ingot disk)
:typicality 1.00)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Domain "House"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(term :name house
:importances ((cube 0.50) (prism 0.50) (ingot 0.48) (pyramid 0.48))
:exemplars (proto-house simple-house)
:relations (((house always has part heya) "heya is always a part of house")
((house always has part yane) "yane is always a part of a house")))

```

```

(term :name      heya
    :reminders  (house 0.67)
    :importances ((cube 0.48) (cylinder 0.48))
    :exemplars  (proto-heya simple-house-heya)
    :relations  ((heya sometimes is inferred from cube) "cube occasionally suggests heya")
                ((heya always is part of house) "heya is always a part of house")))

(term :name      yane
    :reminders  (house 0.70)
    :importances ((prism 0.48) (cone 0.35) (ingot 0.48) (pyramid 0.40))
    :exemplars  (proto-yane house-yane)
    :relations  ((yane sometimes is inferred from pyramid) "pyramid sometimes suggests yane")
                ((yane always is part of house) "yane is always a part of a house")
                ((yane sometimes is inferred from ingot) "ingot sometimes suggests yane")
                ((yane sometimes is inferred from cone) "cone sometimes suggests yane")
                ((yane sometimes is inferred from pyramid) "pyramid sometimes suggests yane")
                ((yane sometimes is inferred from prism) "prism sometimes suggests yane")))

(term :name      simple-house
    :comment    "(an exemplar of a house)"
    :category   house
    :features   (yane heya)
    :typicality 2.00)

(term :name      proto-house
    :comment    "(an exemplar of a house)"
    :category   house
    :features   (proto-yane proto-heya)
    :typicality 2.00)

(term :name      house-yane
    :comment    "(an exemplar of house yane)"
    :category   yane
    :features   (prism)
    :typicality 1.00)

(term :name      proto-yane
    :comment    "(an exemplar of house yane)"
    :category   yane
    :features   (prism)
    :typicality 1.00)

(term :name      proto-heya
    :comment    "(an exemplar of house heya)"
    :category   heya
    :features   (cube)
    :typicality 1.00)

(term :name      simple-house-heya
    :comment    "(an exemplar of simple house heya)"
    :category   heya
    :features   (cube)
    :typicality 1.00)

```