

仮想世界データベースシステムにおける マルチモーダルデータ操作言語の処理系

渡辺 知恵美[†] 増永 良文[‡]

[†] お茶の水女子大学大学院人間文化研究科数理・情報科学専攻

[‡] お茶の水女子大学理学部情報科学科

〒112-8610 東京都文京区大塚 2-1-1

E-mail: chiemi@dblab.is.ocha.ac.jp, masunaga@is.ocha.ac.jp

我々が研究を進めている仮想世界データベースシステム (Virtual World Database System : VWDB) [1,2,3] では, 実世界のオブジェクトを 3 次元的な姿形をそのままにデータベースに格納し, ユーザはデータベースに格納されたオブジェクトを実世界さながらに眺めたり手や声でオブジェクトを操作することができる. 我々は [3] において音声やジェスチャなどを用いたマルチモーダルデータ操作言語を VWDB の操作記述言語として提案した. 現在, VWDB プロトタイプシステムにおいてマルチモーダルデータ操作言語の言語解析系である「マルチモーダルによる更新要求機能」の実装を進めている. 本稿では「マルチモーダルによる更新要求機能」におけるマルチモーダル入力の解析方法およびデータベースへの更新メッセージ生成方法について述べる.

Parsing and Evaluation processor of a Multimodal Interaction Scheme in the Virtual World Database System

Chiemi Watanabe Yoshifumi Masunaga

[†] Graduate School of Humanities and Sciences, Ochanomizu University

[‡] Department of Information Sciences, Faculty of Science, Ochanomizu University

ABSTRACT

We have been developing the “Virtual World Database System (VWDB)”[1,2,3]. The VWDB captures both the real world and the objects in it as possible as they are in the real world. In this system user can see and operate the database objects as if they exist in the real world.

In the previous paper [3], a design of a multimodal interaction scheme in the VWDB is reported. Based on that result, this paper describes an algorithm of parsing and evaluation processor of multimodal inputs. The processor translates a multimodal user interaction to an equivalent message to the backend database objects.

1 はじめに

我々は、リレーショナルデータベースシステムとオブジェクト指向データベースシステムに続く次世代のデータベースとなる仮想世界データベースシステム (Virtual World Database System : VWDB) の設計と実装を進めている [1],[2],[3]。VWDB の設計理念は「仮想世界をデータベースにする (Making Virtual World a Database System)」という言葉に集約される。オブジェクトはデータベース中の仮想世界格納され、ユーザは VR システムを通して仮想世界に没入し実世界しながらに仮想世界を眺めたり手でオブジェクトを動かしたりすることができる。

我々は、VWDB の実現に向けて VWDB データモデルの定義とデータモデルに基づくプロトタイプシステムの実装を行っている。プロトタイプシステムではオブジェクト指向データベースシステムと VR システムとをシステム統合し、バックエンド DB 機能、VR フロントエンド機能として図 1 に示す各機能を実装している。

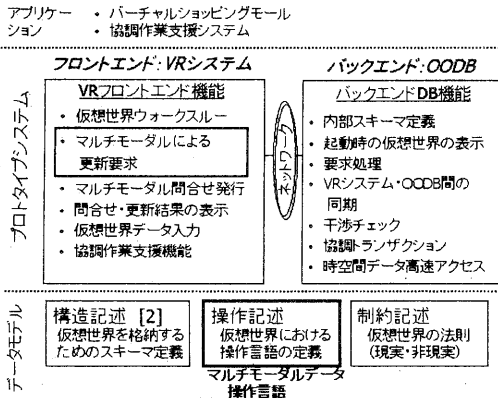


図 1: VWDB の概要図

このうち、我々は [3] において音声やジェスチャを用いたマルチモーダルデータ操作言語を VWDB の操作記述言語として提案した。現在プロトタイプシステムにおいてこの言語を用いた操作を実現するために「マルチモーダルによる更新要求機能」を VR フロントエンドの 1 機能として実装している。この機能は音声やジェスチャなどの入力からマルチモーダルデータ操作言語およびそれが意図するユーザの要求を理解し、データベースへの要求メッセージを作成し送信する。

本稿ではマルチモーダルによる更新要求機能における解析および要求メッセージ生成方法について述べる。

2 マルチモーダルデータ操作言語

我々は [3] において、音声やジェスチャによるマルチモーダルデータ操作を規定するために有限オートマトンを用いたマルチモーダルデータ操作言語記述モデルを提案した。図 2 に示すオートマトン M_{11} は、音声とジェスチャによる移動操作のうち移動対象となるオブジェクトを特定するインタラクションを表わしている。

$$M_{11} = \{Q_{11}, \Sigma_{11}, \delta_{11}, q_0, F_{11}\}$$

$$Q_{11} = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\Sigma_{11} = \{v_1, v_2, v_3, v_{12}, g_1, g_2, g_5, g_6\}$$

$$F_{11} = \{q_4\}$$

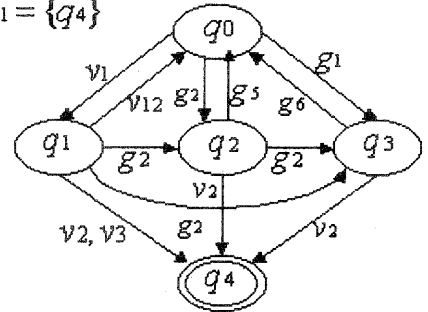


図 2: マルチモーダルによる更新要求機能

図中 q_0, \dots, q_4 はオブジェクトの状態を表す。 q_0 が初期状態、 q_1 は M_{11} の最終状態である。また、 v_i は音声による入力、 g_j はジェスチャによる入力を表す。

オートマトン M_{11} は、例えば「あるオブジェクトを指差して『これ』と発話する」インタラクションを規定する。まず指差しのジェスチャ (g_2) によってオブジェクトは初期状態 (q_0) から移動対象の候補 (q_2) となり、「これ」と発話する (v_2) ことにより指差されたオブジェクトは移動対象として選択 (= 同定) される (q_4)。 M_{11} ではその他にあるオブジェクトに触れて『これ』と発話する、「探索条件を発話し『これ』または『これら』と発話する」インタラクションが規定されている。

M_{11} は移動操作のうちの一部のインタラクションをあらわしているが、移動操作のためのマルチモーダル操作言語を設計するためには、オブジェクトの同定から移動終了までの一連のインタラクションを規定する必要がある。

一連の移動操作におけるオブジェクトの状態は、「初期状態」から移動操作の対象として選ばれている「移動できる状態」、実際に移動している「移動中の状態」、移動が完了した「最終状態」と大きく 4 状態に分けることができ、これらのオブジェクトの状態に基づいて移動操作を以下の 3 ステージに分けることができる。

- ・ ステージ 1 (S_1): 初期状態から移動できる状態まで、つまり移動対象となるオブジェクトを同定

する。

- ステージ 2 (S_2): 移動できる状態から移動中の状態まで、移動操作のための各設定(方向、速度など)を決めて移動を開始させる。
- ステージ 3 (S_3): 移動中の状態から最終状態まで、つまり移動中のオブジェクトを停止させる。

一連の移動操作を表すオートマトンは、各ステージにおけるインタラクションを規定するオートマトンを設計し、それらを結合することによって得ることができる。このようにして作成した 12 種類の発話と 7 種類のジェスチャによる移動操作を規定するオートマトン $M = (Q, \Sigma, \delta, q_0, q_{16})$ を図 3 に示す。

マルチモーダルデータ操作言語は、オートマトン M が受理される入力系列の集合 $L(M)$ である。

以下に移動操作のインタラクション例と、それを表す入力系列 $l \in L(M)$ を示す。

- 「(オブジェクト A を指差して)『これ』を『前』へ『移動』」といてオブジェクトの移動を開始させ、『止まれ』と発話して停止させる。
- 入力系列 ($l \in L(M)$)
 $l = g_2 v_2 v_7 v_{10} v_{11}$
- オートマトン M における状態遷移の様子
 1. $\delta(q_0, g_2) = q_2$
 2. $\delta(q_2, v_2) = q_4$
 3. $\delta(q_4, v_7) = q_8$
 4. $\delta(q_8, v_{10}) = q_5$
 5. $\delta(q_5, v_{11}) = q_{16}$

3 マルチモーダルによる更新要求機能

マルチモーダルによる更新要求機能における処理の流れを図 4 に示す。

本機能はマルチモーダル入力部 (= module) とマルチモーダル入力解析部に分けられ、マルチモーダル入力解析部は構文・意味解析部と要求メッセージ生成部によって構成される。構文・意味解析方法は自然言語処理による解析手順を参考にし、インタラクションの意味表現には意味ネットワークを用いた。

3.1 マルチモーダル入力部

プロトタイプシステムでは入力モードとして音声とジェスチャを用いている。音声は音声入力装置および音声認識部、ジェスチャはデータグローブおよびジェスチャ認識部を通して入力される。

手の形状はデータグローブによって計測され、ジェスチャ認識部に送られる。ジェスチャ認識部は計測データからジェスチャを認識し、オートマトン M におけるジェスチャ入力記号 g_i と g_i のパラメタ集合をマルチモーダル入力解析部に送る。 g_i のパラメタとはデータベースへの要求メッセージを作成するために必要となる情報で、以下のパラメタがある。

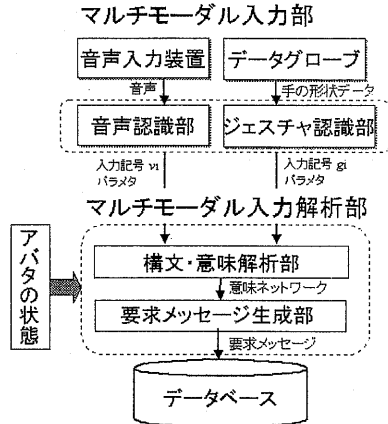


図 4: マルチモーダルによる更新要求機能

- 入力が行われた時区間 $time.int = [t_s, t_e)$ 入力の開始時刻 t_s 、終了時刻 t_e ($t_s < t_e$) からなる半开区間を入力が行われた時区間とする。
- 入力記号 g_i の詳細な情報を表すパラメタ。以降これを入力パラメタと呼ぶ。例えば「指差す」(g_2) ジェスチャの入力パラメタは以下の通りである。
 - $dir = \{(dx, dy, dz, t_s), \dots, (dx, dy, dz, t_e)\}$: 時刻 t_i ($t_s \leq t_i < t_e$) における人差し指の方向ベクトルの集合¹
 - $pos = \{(x, y, z, t_s), \dots, (x, y, z, t_e)\}$: 時刻 t_i における人差し指の指先の位置座標の集合
 - $obj = \{(oid, t_s), \dots, (oid, t_e)\}$: 時刻 t_i に人差し指で指差されているオブジェクトの識別子

音声もジェスチャと同様に、音声入力装置で計測して音声認識部で発話内容を認識し、音声入力記号 v_i と v_i のパラメタ集合をマルチモーダル入力解析部に渡す。

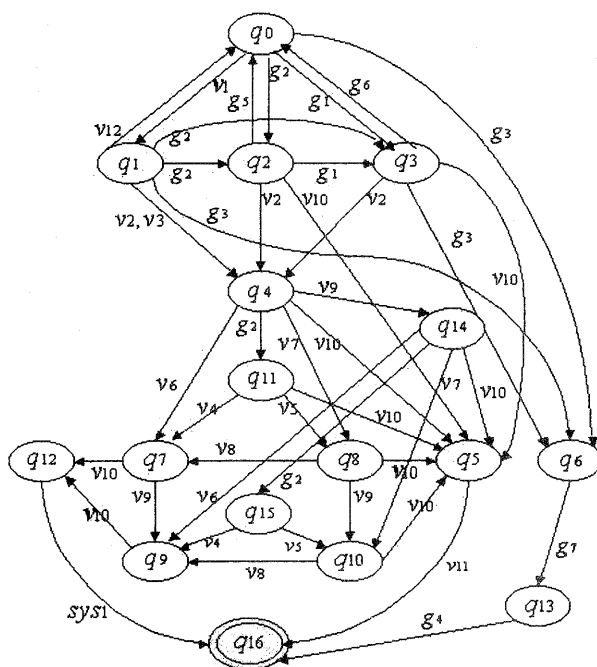
3.2 マルチモーダル入力解析部

マルチモーダル入力解析部は解析部と要求メッセージ生成部によって構成される。解析部は音声・ジェスチャ認識部から受け取った入力を解析し、入力系列の構造を意味ネットワーク表現で記述する。要求メッセージ生成部は解析部で生成された意味ネットワークから要求メッセージを生成しデータベースに送信する。

3.2.1 要求メッセージ生成のタイミング

マルチモーダル入力解析部を設計するためには要求メッセージ生成のタイミングについて考慮する必要がある。ユーザはマルチモーダルによる操作においてデータベース中のオブジェクトと対話的に操作を進める。例えば、

¹ 方向ベクトルは指先の位置座標と第一関節の位置座標によって求められる



$$M = (Q, \Sigma, \delta, q_0, F)$$

$$Q = \{q_1, \dots, q_{16}\}, \Sigma = \{v_1, \dots, v_{12}, g_1, \dots, g_7\}, F = \{q_{16}\}$$

Q : 状態

- q0:初期状態
- q1:操作対象の候補になった状態
- q2:操作対象の候補になった and
- q3:操作対象の候補になった and 接触状態
- q4:移動できる状態
- q5:移動中(停止の合図があるまで直線的に移動しつづける)
- q6:移動できる状態 and g7の入力待ち状態
- q7:移動できる状態 and 移動先が設定された状態
- q8:移動できる状態 and 移動の方向が設定された状態
- q9:移動できる状態 and 移動先と速度が設定された状態
- q10:移動できる状態 and 移動の方向と速度が設定された状態
- q11:移動できる状態 and v4またはv5の入力待ち状態
- q12:移動中(移動先まで直線的に移動する)
- q13:移動中(ユーザの手について移動する)
- q14:移動できる状態 and 移動速度が設定された状態
- q15:移動できる状態 and 速度が設定された状態 and v4またはv5の入力待ち状態
- q16:終了状態

SYS : システム

sys1 移動の終了(目的地に到着した)

v : 音声

- v1: 探索条件
- v2: 「これ」or「それ」(単数)
- v3: 「これら」or「それら」(複数)
- v4: 「ここ」or「そこ」(位置を決定)
- v5: 「こっち」or「そっち」or「あっち」
(方向を決定)
- v6: 座標値
- v7: 方向(前、後、右、左、上、下)
- v8: 距離
- v9: 速度
- v10: 「進め」or「移動」
- v11: 「止まれ」
- v12: 探索条件から外れる

g : ジェスチャー

- g1: 手が触れている
- g2: 指差し
- g3: 掴む
- g4: 放す
- g5: 指差した先から外れる
- g6: 手から離れる
- g7: 手の位置が動く

図 3: 初期状態から移動が終了するまでを表したオートマトン

第2章で挙げた移動操作のインタラクション例において、ユーザはオブジェクトと以下のような対話を行う。

- (a) オブジェクトの同定 (指差して「これ」と発話する) → 指されたオブジェクトが光る
- (b) 移動の開始 (「前」に「移動」と発話する) → オブジェクトが移動を開始する
- (c) 移動の終了 (「止まれ」と発話する) → オブジェクトが停止する。

このような対話的な操作を実現するためには、マルチモーダル入力解析部は移動操作のための全ての入力を受け取ってから要求メッセージを生成するのではなく、入力系列が (a)(b)(c) の要求をした時点で要求メッセージを生成していく必要がある。

図5に入力系列 $l = g_2 v_2 v_7 v_{10} v_{11}$ においてマルチモーダル入力解析部が受け取る入力とそれに伴うオートマトン M の状態遷移、入力系列および入力系列が (a)(b)(c) の要求を示すタイミングをまとめた。

入力	状態	ステージ	入力系列	要求メッセージ
	q_0			
g_2	q_2	S_1	g_2	
v_2	q_4		$g_2 v_2$	(a) オブジェクト同定
v_7	q_8	S_2	$g_2 v_2 v_7$	
v_{10}	q_5		$g_2 v_2 v_7 v_{10}$	(b) 移動開始
v_{11}	q_{16}	S_3	$g_2 v_2 v_7 v_{10} v_{11}$	(c) 移動終了

図5: 入力に伴う状態遷移と要求を示すタイミング

図5から以下のことがわかる。

- (a) は入力系列が $g_2 v_2$ (g_2 :「指差す」, v_2 :「これ」) となった時点で生成できる。このときのオートマトンの状態は q_4 であり、 S_1 の最終状態である。
- (b) は入力系列が $g_2 v_2 v_7 v_{10}$ (v_7 :「前」, v_{10} :「移動」) となった時点で生成できる。このときのオートマトンの状態は q_5 であり、 S_2 の最終状態である。
- (c) は入力系列が $g_2 v_2 v_7 v_{10} v_{11}$ (v_{11} :「止まれ」) となった時点で生成できる。このときのオートマトンの状態は q_{16} であり、 S_3 およびオートマトン M 全体の最終状態である。

このことから、データベースに送信する要求はオートマトンの状態が各ステージ S_1, S_2, S_3 の最終状態となったときに生成すればよいことがわかる。

そこで、オートマトン M の状態のうち各ステージの最終状態を M の準最終状態 q_{fi} であるとし、 M の

状態が q_{fi} となった時点で要求メッセージ生成を行うこととする。

3.2.2 構文・意味解析部

構文・意味解析部はオートマトン M を用い以下の手順で処理を行う。

1. 入力記号 g_i (または v_i) を受け取るとオートマトン M に従って状態を遷移させる。
2. オートマトン M から得られる正規文法を用いて入力系列の構文を表す解析木を生成する。図6(a)は入力記号 v_{10} を受け取った時点の解析木である。これと同時に解析木に対応した意味ネットワーク(図6(b))を生成する。
3. オートマトンの状態が準最終状態 q_{fi} であれば、意味ネットワークを要求メッセージ生成部に渡す。

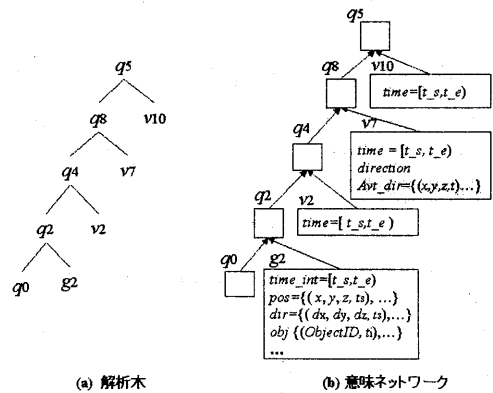


図6: v_{10} 入力時の解析木と意味ネットワーク

意味ネットワークは解析木と同じ構造をもつ。各ノードは属性集合 $attr(g_i)$ (または $attr(v_i)$) を持ち、その値は入力系列と共に音声・ジェスチャ認識部から送られるパラメタ集合によって求められる。例えば、ノード g_2 の属性集合 $attr(g_2)$ は

$$attr(g_2) = \{time_int, dir, pos, obj\}$$

- $time_int$: 入力が行われた時間,
- $dir = \{(dx_s, dy_s, dz_s, t_s), \dots, (dx_e, dy_e, dz_e, t_e)\}$: 時刻 $t_i (t_s \leq t_i < t_e)$ における人差し指の方向ベクトルの集合²
- $pos = \{(x_s, y_s, z_s, t_s), \dots, (x_e, y_e, z_e, t_e)\}$: 時刻 t_i における人差し指の指先の位置座標の集合
- $obj = \{(oid_s, t_s), \dots, (oid_e, t_e)\}$: 時刻 t_i に入差し指で指差されているオブジェクトの識別子の集合

であり、属性値は入力 g_2 のパラメタ群 (第3.1節の例を参照) によって求められる。

²方向ベクトルは指先の位置座標と第一関節の位置座標によって求められる

また v_7 (「前」「右」「左」など…) のようにアバタの状態が必要となる入力ノードには、アバタの状態情報が属性として付加される。

3.2.3 要求メッセージ生成部

要求メッセージ生成部は解析部によって作られた意味ネットワークを用いてコード生成を行う。意味ネットワークのうち状態 q_i に対応するノードは評価メソッド (Evaluation Method) $EM(q_i)$ をもつ。 $EM(q_i)$ は q_i にリンクしている2つのノード q_j, v_k (または g_k) の属性値をもとに q_i 自身の属性値を求めるメソッドである。要求メッセージ生成部は $EM(q_i)$ を最左下のノードから最上位のノードまで順に実行させる。最上位のノードは準最終状態 q_f 、または最終状態 q_F のノードである。これらのノードは要求メッセージを表す属性 msg を持ち、 $EM(q_f)$ を実行することによって要求メッセージ msg が求められる。こうして得られた msg をデータベースに送信する。

4 要求メッセージ生成例

要求メッセージ生成例として、移動操作のインタラクション例と要求メッセージ生成の流れを示す。

4.1 インタラクション例

有限オートマトン M で規定される移動操作インタラクション例を以下に示す。

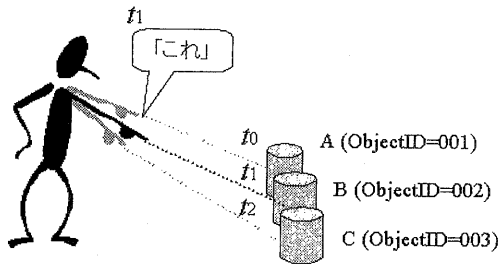


図 7: 指差して「これ」

- 時刻 t_0 のとき、指差しのジェスチャ (g_2) を始める。
このとき指差しているオブジェクトは A であり、そのオブジェクト ID は 001 である。
- 時刻 t_1 のとき、「これ」と発話する (v_2)。
このとき指差しているオブジェクトは B であり、そのオブジェクト ID は 002 である。
- 時刻 t_2 のとき、指差しのジェスチャをやめる。
このとき指差しているオブジェクトは C であり、

そのオブジェクト ID は 003 である。

- 時刻 t_3 のとき、「前」と発話する。
このときアバタの向きベクトルは (1, 0, 0) である。
- 時刻 t_4 のとき、「移動」と発話する。
- 時刻 t_5 のとき、「止まれ」と発話する。

4.2 処理手順

(a) オブジェクト同定までの処理

マルチモーダル入力部

時刻 t_0 から時刻 t_2 にかけてジェスチャ認識部が g_2 を認識、時刻 t_1 に音声認識部が v_2 を認識し、入力記号とそのパラメータをマルチモーダル入力解析部に送る。

構文・意味解析部

まず g_2 を受け取ることによってオートマトン M の状態は q_0 から q_2 に遷移し、 v_2 を受け取ることによって q_1 に遷移する。これと同時に解析木と意味ネットワークが生成される。

$q_{f_1} = q_1$ であるため、オートマトン M の状態が q_1 となった時点で意味ネットワークがメッセージ生成部へ渡される。この時点の意味ネットワークを図 8(a) に示す。

g_2 の属性 $attr(g_2)$ と v_2 の属性 $attr(v_2)$ 、およびこれらの属性値はそれぞれ以下の通りである。

ノード g_2 :

- $attr(g_2) = \{time_int, pos, dir, obj\}$
(各属性の説明は第 3.2.2 節の例を参照)
- 属性値:
- $time_int = [t_0, t_2 + 1)$
- pos, dir, obj は $t_i (t_0 \leq t_i < t_2)$ におけるそれぞれの値が g_2 の入力パラメータから求められる。特に obj は時刻 t_1, t_2, t_3 において $obj = \{(001, t_0), \dots, (002, t_1), \dots, (003, t_2)\}$ である。

ノード v_2 :

- $attr(v_2) = \{time_int\}$
- 属性値: $time_int = [t_1, t_1 + 1)$

要求メッセージ生成部

解析部から受け取った意味ネットワークを用い、 q_0, q_2, q_1 の順で評価メソッド $EM(q_0), EM(q_2), EM(q_1)$ を実行する。ノード q_2, q_1 における属性 $attr(q_2), attr(q_1)$ および評価メソッド $EM(q_2), EM(q_1)$ の行う処理を以下に示す。なお、ノード q_0 は属性はなく評価メソッド $EM(q_0)$ は何もしない。

ノード q_2 :

- $attr(q_2) = \{time_int, obj\}$
- $time_int = [t_s, t_e + 1)$: q_2 である時間を開始時刻 t_s と終了時刻 t_e で表す。

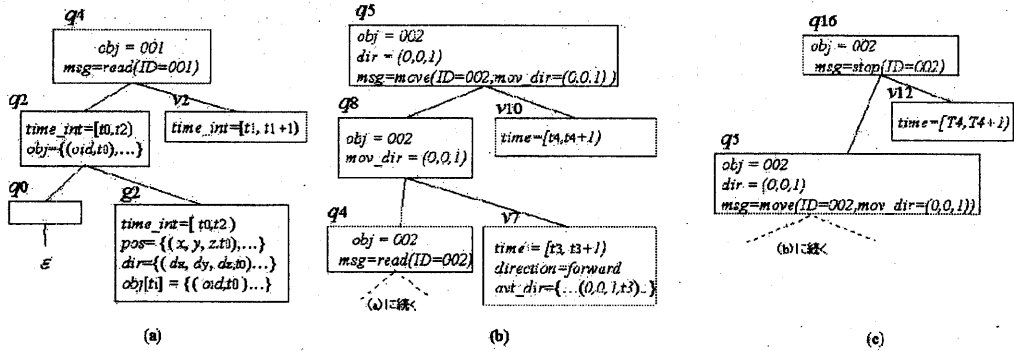


図 8: 要求メッセージ生成部に渡される意味ネットワーク

- $obj = \{(t_s, obj_s), \dots, (t_e - 1, obj_e)\}$: $time_int$ で表される時区間において移動対象の候補となっているオブジェクトの識別子の集合。
- $EM(q_2)$ における処理:
 g_2 の属性値から $time_int = [t_0, t_2 + 1)$, $obj = \{(001, t_0), \dots, (003, t_2)\}$ を取得し, q_2 の属性 obj の属性値とする。

ノード q_4 :

- $attr(q_4) = \{obj, msg\}$:
 obj : 移動対象となるオブジェクトの識別子
 msg : 更新要求メッセージ
- $EM(q_4)$ における処理:
 v_2 の属性値 $time = [t_1, t_1 + 1)$ と q_2 の属性値 obj を参照する。 obj において時刻 t_1 におけるオブジェクト識別子 $(001, t_1)$ であることから q_4 の属性 obj の値を $obj = 001$ とする。 また, $obj = 001$ から要求メッセージ $msg = read(ID = 001)$ を生成しデータベースに送信する。

(b) 移動開始までの処理

マルチモーダル入力部

音声認識部が時刻 t_3 に v_7 を, 時刻 t_4 に v_{10} を認識し, 入力記号とパラメータ群をマルチモーダル入力解析部に送る。

構文・意味解析部

v_7 を受け取ることによってオートマトンの状態は q_4 から q_8 に遷移, v_{10} を受け取ることによって $q_5 = q_{f_2}$ に遷移し, 図 8(b) に示す意味ネットワークをメッセージ生成部へ渡す。このときのノード v_7, v_{10} の属性および属性値は以下のとおりである。

ノード v_7 :

- $attr(v_7) = \{time_int, direction, avt_dir\}$
- $direction \in \{forward, backward, right, left, up, down\}$: 移動方向
- $avt_dir = \{(dx_s, dy_s, dz_s, t_s), \dots, (dx_e, dy_e, dz_e, t_e)\}$: 時刻 $t_i (t_s \leq t_i < t_e)$ におけるアバタの向きベクトル

属性値:

- $time_int = [t_3, t_3 + 1)$
- $direction = forward$
- $avt_dir = (1, 0, 0, t_3)$

ノード v_{10} :

- $attr(v_{10}) = \{time_int\}$
- $attr(v_{10})$ の属性値: $time_int = [t_4, t_4 + 1)$

要求メッセージ生成部

図 8(b) の意味ネットワークを受け取り, q_8, q_5 の評価メソッド $EM(q_8), EM(q_5)$ を実行する。ノード q_8, q_5 における属性 $attr(q_8), attr(q_5)$ および評価メソッド $EM(q_8), EM(q_5)$ の行う処理を以下に示す。

ノード q_8 :

- $attr(q_8) = \{obj, mov_dir\}$
- obj : 移動するオブジェクトの識別子。
- $mov_dir = (dx, dy, dz)$: 移動方向を表すベクトル
- $EM(q_8)$ における処理:
 ノード v_7 の属性値 $direction = forward, avt_dir = \{... (1, 0, 0, t_3)\}$ より, オブジェクトの移動方向を示す方向ベクトル $mov_dir = (1, 0, 0)$ を求めノード q_8 の属性値とする。ノード q_4 の属性値 $oid = 001$ を q_8 の属性値とする。

ノード q_5

- ・ $attr(q_5) = \{obj, dir\}$:
- obj : 移動するオブジェクト識別子.
- $dir = (dx, dy, dz)$: 移動方向を表すベクトル
- ・ $EM(q_5)$ における処理:
ノード q_8 の全ての属性値をノード q_5 の属性値とする. これらの値から要求メッセージ $msg = move(ID = 002, mov.dir = (1, 0, 0))$ を生成し, データベースに送信する.

(c) 移動終了までの処理

マルチモーダル入力部は時刻 t_5 に v_{11} を認識する. 構文・意味解析部は v_{11} を受け取ることによって $q_{16} = q_F$ に遷移し図 8(c) に示す意味ネットワークをメッセージ生成部へ渡す. このときノード v_{11} の属性 $time_int$ の値は $time_int = [t_5, t_5 + 1)$ である. 要求メッセージ部は $EM(q_{16})$ を実行することにより属性値 $obj = 001$ を求め, これを用いて要求メッセージ $msg = stop(ID = 002)$ を生成してデータベースに送信する.

5 関連研究

マルチモーダルユーザインタフェース (Multimodal User Interface: MUI) における研究は 1980 年の "put that there" プロジェクト [4] が先駆けと考えられる. その後国内外で数多くの研究が行われているが, [5][6][7][8]. そのほとんどは MUI を用いた応用システムに着目したものである. しかし, マルチモーダル用言語の記述形式や汎用的な MUI に対応した入力統合・解析方式に関する研究も [5][6] において進められている. [6] は自然言語の文法記述形式である DCG (Definite Clause Grammar) を拡張したマルチモーダル用言語文法記述形式 MM-DCG を提案している. また [5] は自然言語の構文解析技術を応用したマルチモーダル統合・解析の枠組みを提案している. これらの研究と比較して VWDB における MUI のオートマトンによるマルチモーダル言語記述方式およびその解析系は以下の特徴を持つ.

- (1) 音声とそれ以外の入力モードを同等に扱う
- (2) アバタの状態を考慮に入れた操作が可能
- (3) 一連の操作において対話性がある

(1) に関しては, [5] は音声による自然言語を主とし, ジェスチャは指示代名詞を補完する程度の補助的な役割としている. 一方 [6] および VWDB では全ての入力モードを同等に扱っており, 例えば手でオブジェクトをつかんで動かすなどの音声以外のモードで操作を行うことが可能となる.

(2) は VWDB が 3 次元空間に没入した状態で操作を行うために必要とされる特徴で, 解析時にアバタの状態パラメータを取り入れることでこれを可能としている. 一方, [5], [6] は共に GUI をベースにした 2 次元のシステムを対象としておりシステム全体を鳥瞰的に眺めているため, アバタの状態は考慮に入らない.

(3) は VWDB の最も大きな特徴であり第 3.2.1 節で議論した項目を指す. [5], [6] は移動操作における全ての入力を受け取ってから解析および要求メッセージ生成を行う. 一方, VWDB はデータベース中のオブジェクトとの対話的な操作を可能にするため一連の移動操作において複数回メッセージ送信を行う. なお今回移動操作のための要求メッセージを送信するタイミングとしてステージ S_1, S_2, S_3 を利用したが, さらにこれらをトランザクション管理に利用することができる. 具体的には, 入れ子トランザクション [10] において, この S_1, S_2, S_3 をサブトランザクションの単位として有効に扱うことができる.

6 おわりに

VWDB におけるマルチモーダル操作言語をプロトタイプシステムで実現するための「マルチモーダルによる更新要求機能」における解析および更新メッセージ生成方法について述べた. 更新メッセージを送信する単位として移動操作オートマトン M の設計で用いたステージ S を利用し, 音声またはジェスチャ入力に付加したパラメータを属性にもつ意味ネットワークにより更新メッセージを生成した.

現在, この機能を実装中である. 実装および評価にあたって考慮すべき点としては音声およびジェスチャ認識部における認識率の問題がある. これについては今後実験すると共に, 認識率が低い場合は認識部で複数の候補を挙げオートマトンの経路を基に候補を絞り込むなどの対策を考える必要がある.

また第 5 章で触れたトランザクション管理についても今後取り組むべき課題である.

参考文献

- [1] 渡辺知恵美, 増永良文: 仮想環境データベースシステムのためのデータモデル. 情報処理学会データベースシステム研究会報告, No.119, Jul. 1999, pp. 159 - 164(1999).
- [2] Masunaga, Y. and Wanatabe, C.: The Virtual World Database System - Its Concept, Design and Prototyping -, Advances in Multimedia and Databases for the New Century, A Swiss/Japanese Perspective, pp.61 - 70, World Scientific(2000).
- [3] 渡辺知恵美, 増永良文: 仮想世界データベースシステムにおけるマルチモーダルデータ操作言語の設計, 第 11 回データ工学ワークショップ, 7B-5, 6p(2000).
- [4] Bolt R. A.: Put that there: Voice and Gesture at the Graphics Interface, Computer Graphics, 14, 3(1980).
- [5] Koons, D.B., Spaarrell, C.J. and Thorisson, K.R.: Integrating simultaneous input from speech, gaze, and hand gestures, Intelligent Multimedia Interfaces, MIT Press, pp.257 - 276(1993).
- [6] 島津秀雄, 高島洋典: マルチモーダル Definite Clause Grammar (MM-DCG), 電子情報通信学会論文誌, D-II, Vol. J77-D-II, No.8, pp.1438 - 1446(1994).
- [7] Cohen, P.R.: Natural language techniques for multimodal interaction, 情報通信学会論文誌, D-II, Vol. J77-D-II, No.8, pp.1403 - 1416(1994).
- [8] 長尾 確: インタラクティブな環境を作る, 共立出版 (1996).
- [9] 田村博 (編): ヒューマンインタフェース, オーム社 (1999).
- [10] Beeri C., Bernstein P. A. and N. Goodman: A Model for Concurrency in Nested Transaction Systems, Journal of the ACM, Vol.36, No.2, pp.230 - 269 (1989).