

## 日本語アニメシナリオからの意味抽出

吉川耕平 舟渡信彦 花田恵太郎 宮本雅之

シャープ株式会社 技術本部 コンピュータシステム研究所

日本語で記述されたアニメシナリオから実時間のアニメーションを生成するシステムを目指した研究を進めている。本稿ではアニメーションを登場物、登場物による動作、シナリオの記述の3つからなると考える。このうち登場物については画面への表示形態と、各種の属性、他の登場物との関係を用いて表現する。動作については登場物の移動、向きの変化、属性の変更、関係の変化、他の登場物の動作のきっかけを与えること、の5つと考える。これらをシナリオの持つ意味として、日本語シナリオにおける時間経過を前提とした意味解析の方法とその表現形式について述べる。

### A Semantic Representation of Natural Language Animation Scenario

Kouhei YOSHIKAWA, Nobuhiko FUNATO, Keitaro HANADA, Masayuki MIYAMOTO  
Computer Systems Laboratories, Sharp Corp., Corporate R&D Group.  
2613-1, Ichinomoto-cho, Tenri-shi, Nara 632, Japan

*abstracts* : A semantic representation for Japanese animation description is introduced. An animation is represented with three notions, actors, their actions, and scenario. An actor has some attributes, and relations between other actors. An action is constructed by five primitives, such as actor's move, its turn of direction, changes of attributes' values, changes of relation, and triggers to other actor's actions. The scenario is represented as a list of actions.

## 1. はじめに

日本語で記述されたシナリオから実時間のアニメーションを生成することを目的に研究を進めている。システムは、ユーザによるアニメーションの作成系、画像化に必要な詳細データを作る詳細データ作成系、アニメーション実行系の三つからなる(図1)。アニメーション作成系ではアニメーションの登場物の表示形態データ(背景を含む)とシナリオ記述解釈のための動作概念辞書データ、及びそれらを使って展開するシナリオを日本語により作成する。詳細データ作成系は、ユーザが子細に指定する必要がないデータを中心に推論を行い、画像化データを作る。実行系では、データ作成系から送られたデータをもとに後に述べるアニメーション実行モデルに基づき実時間で処理し、表示を行う。

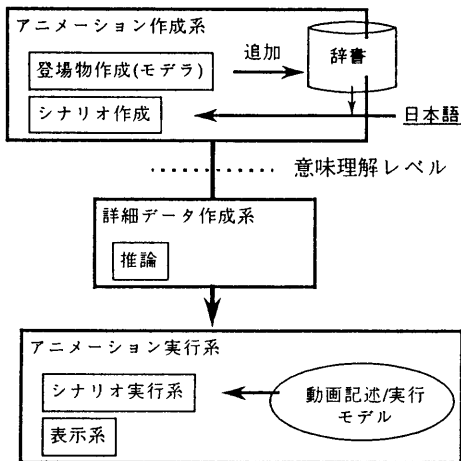


図1.

本稿では、このうちアニメーション作成系における日本語シナリオ解釈の方式と、解釈のための動作概念辞書の内容について述べ、図1.における意味理解レベルの表現を導く。この表現を用いてシナリオの内容に関する簡単なQ&Aに答えることができる。

一般にアニメーションを作成するためには莫大なデータが必要である。ここではその作成の手間を少しでも省くため、

1. 動作の概念
2. アニメーション登場物と、それが可能な動作
3. シナリオ記述

の3つを分離して作成/管理し、それぞれの再利用性を高める。また上記1, 2.の整備により日本語からアニメーションへのメディア間変換を目指す。

以下では日本語シナリオ文、シナリオで表現できる動作、我々が前提としているアニメーション記述/実行系について簡単に説明した後、アクタ(アニメーション登場物)の表現、アクタによる動作の概

念を解説し、最後にそれらを用いる日本語シナリオ解釈の方式およびその解釈結果の表現について述べていく。

## 2. 入力文

シナリオを日本語で記述することで、指示語の利用や繰り返し部分の省略などが可能である。このメリットを生かすため、それらが指し示す内容について、入力が時間の経過に沿った複数の文の並びであることを前提にして、後で述べる方法で固定する。また、動作概念として動作実行のための前提条件が辞書に記述されている場合は、それを満たすための動作を自動的に生成することも可能にする。

ここでは現在開発中のシステムでの、入力文に対する制約について述べる。

・文は単文とし、動詞が二つ以上のものは扱わない。

▶「ベッドから降りて、棚の前まで歩く」

・動作内容として、主体や対象に不特定多数を指すものや複数の主体による協調的な動作は考えない。

▶「みんなが来る」

▶「ムーミンとノンノンが木を倒す」

・入力される文の順序はおおむね動作の起こる順番に対応するものとする。

時間幅を持つ動作で、先に始まっても先に終わらない場合や、後から述べても実は同時に始まるという場合は、後節で述べる時間関係の処理によって解釈する。特に動作の最中であることを述べたり、動作の開始や終了があったことを述べたりするアスペクトの表現「～し始める」「～している」「～終わる」などを扱えるものとする。

・時間関係を表現する以下のような接続詞についても扱えるものとする。

続いて、同時に、しばらくして、その間に、

・他動詞は、対象物の自動詞的な動作が定義されていることを前提として、その開始のトリガを与えるものと解釈する。

▶「人が本を開く」

= 「人が本にトリガを与える」+ 「本が開く」  
なお他動詞でも、再帰的な動作は別の方法で扱う。

▶「人が目を開く」

この詳細については動作概念の表現で触れる。

## 3. シナリオ解釈の範囲

日本語によるシナリオ記述では、ユーザが当然すぎて書かないものや、一見矛盾するような内容、仮定的な内容などが書かれることが予想される。アニメーション作成系ではこのような常識に関する推論や、矛盾の抽出には重点を置かない。すなわち書かれたことはすべて真実であり、また書か

れなかったことは画面表示に不必要なものである、という前提で解析を行う。

システムの目的がアニメーション作成であるため、文は視覚的な動作として表現可能な内容を述べているものと解釈する。現在のところ、そのような動作の中から、特にアニメーション記述のため最も基本的と考えられる動詞[5,6]約60個(後述)によって記述できる動作を対象に解析を進めている。

アニメーション作成系でのシナリオ解釈の範囲について例を挙げて示す。

1. 例えばベッドと窓についての記述があっても壁や床があるとは認識しない。このような常識に関する知識は詳細データ作成系で対処すると考えている。
2. 「ドアを開ける」という文章は、必ず結果としてドアが開くものとする。ただし「ベッドに向かって歩き始める」というような時間幅を持つ動作の開始の記述の場合は必ずしもベッドに到着しなくてもよい。
3. 登場物の動作は、その動作による結果で表現し、また動作の前提は必ず満たされているものと解釈する。例えば「人がドアを開ける」という文が来た場合、陽に移動に関する記述はなくとも、この直前に人がドアまで移動し、かつドアは閉まっていたと解釈する。
4. シナリオ全体の解析結果は、個々の動作の間の開始や終了のタイミングの前後関係、同時関係を表すもので、具体的に例えば何秒かかるか、というような数字までは作らない。
6. 時間経過を表す接続詞「しばらくして」などの意味は、ある程度の時間が経過する、という表現にとどめ、具体的な値までは決定しない。
7. 同じ動詞を使う文でも、特定の格の有無により動作としての意味が異なる。例えば  
「本棚の前まで歩く」  
「部屋の中を歩く」  
の二つを比べると、前者は歩き終わることまでもこの文に含まれているのに対し後者はいつまで歩き続けるのかこの文だけでは分からない。このような場合は、あとに続く文の解析を待って動作の全体(この場合はいつまで歩くか)を決定する。次項参照。
8. 例えば「歩き始める」という動作およびその実行状態に対して「止まる」という動作がその終了を指示すると解釈すべきであるように、アニメーション内の一つの動作を、二つ以上の文で表現することがある。登場物の移動や登場物が持つ属性に注目して動作概念を作ることにより、このような解釈を可能にする。

9. アニメーションの中の世界に上下左右などの方向は考えるが、重力など場の力は考えない。例えば、持っているものを「放し」ても「落ちる」と言われなければ落ちない。

10. 「それが窓ならば開ける」というような仮定的な文、またはアニメーション実行時まで確定できない内容は解釈しない。

当面、文の内容としては日常的な生活の場、例えば部屋の中や家の回りでのストーリーに関するもので、具体的には以下のような文章を考えている。

1. ムーミンが寝ている。
2. ムーミンが目を開ける。
3. ベッドから降りる。
4. たなの方に向かって歩き始める。
5. 机の上の本を持つ。
6. いすの前で止まる。
7. しばらくしていすにすわる。
8. 本を開ける。
9. ノンノンが入ってくる。
10. 部屋の中を歩いている
11. ドアに向かって走る。

#### 4. アニメーション記述/実行モデル

ここでは我々が考えているアニメーション実行系における記述/実行モデルについてその概要を説明する[7]。当モデルは複数の登場物が時間軸に沿ってイベント駆動の動作を行う系を実時間で画面に表示するためのものである。大きく以下の3つの概念を用いる。

- ・アクタ
  - アニメーションの登場物。
  - 属性と、動きのルール(イベント、デマンドの発信を含む)を持つ。
  - 特殊アクタとしてウォッチャ、シナリオ
- ・プロセス
  - アクタが持つ動きのルールに従って実行されている実際の動作。
- ・イベント及びデマンド
  - プロセスの起動あるいは終了のトリガ。
  - デマンドは相手先を指定した動作要求。
  - イベントは相手を特定しない自己内部状態の発信。

アクタは、他のアクタまたはシナリオと呼ぶストーリー制御アクタからデマンドを受けて、自分の持つ動作ルールをもとにプロセスとして動作を開始したり、また実行中のプロセスを終了したりする。ウォッチャアクタは他のアクタ間の相互関係やアクタ内の状態の監視を行い、ストーリー展開のきっかけとなる事象が起こったときにそれをシナリオアクタにイベントとして通知する。

一つのアクタは、一つのプロセスについて、実行中であるか、そうでないかの2つの「状態」を持

つ。ただし、異なる種類のプロセスは同時に何個実行してもかまわない。

このモデルをもとに、アクタによる動作を以下のように考える。

「動作」とは、何らかのプロセスと、その開始/終了の時刻を含めた時区間の全体である。

Action := (St, Et, Cnt)  
 St: Start point of time(開始時点)  
 Et: End point of time(終了時点)  
 Cnt: Contents of action(動作内容)

これを模式的に表したものを図2. に示す。

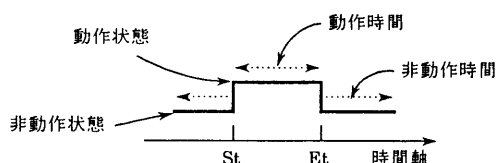


図2.

図において「動作状態」とは、プロセスが実行されている間を指し、その区間を「動作時間」と呼ぶ。プロセスの内容によっては一瞬で終了する動作もある。

以上のモデルを参考にして、日本語シナリオから抽出する意味の内容とその表現について考える。

先にも述べたようにアニメーション作成には次の三つの情報が必要である。

1. 動作の概念
2. アクタ(登場物)と、その動作及び関係
3. シナリオ記述

以下の節では、この各々に関してその持つ意味と表現形式をいくつかの例を挙げて説明する。

## 5. 動作概念

画面に表示できるアクタの動作の基本を、

- a) 移動
- b) 表示や向きに関する動作
- c) 形態の変化
- d) 他のアクタとの関係の変化
- e) 他のアクタに対する動作

の5つとする。

具体的には以下の動詞で表現されるような動作を指す。

### a) 移動

上がる 歩く 行く 動く 降りる 落ちる 来る 着く 下がる 滑る 出る 飛ぶ 止まる 登る 乗る 入る 走る

### b) 表示や向きに関する動作

起きる 傾く 消える 転ぶ 倒れる 立つ 寝る 回る 見る 向く 揺れる

### c) 形態の変化

開く 閉じる 閉まる

### d) 他のアクタとの関係の変化

対象物の移動があるもの  
 受ける 置く かかえる つかむ 持つ  
 対象物には何の変化もないもの  
 指す 触る すわる 近づく 握る 放す

### e) 他のアクタに対する動作

対象物の移動  
 上げる 入れる 動かす 落とす 下ろす ける  
 下げる 出す 飛ばす 止める 投げる  
 対象物の向きの変化  
 起こす 傾ける 転がす 倒す 立てる 回す  
 対象物の形態を変えるもの  
 開ける 閉める 閉じる

a)の移動に関しては、(アクタ、始点、終点)を用いて表現する。ただし、方向やスピードなどの特徴(例えば「上がる/走る」等)が必要なものは、その特徴を合わせて表記し、また領域の制約がある場合はそれをwithやinを付け加えて表現する。

b), c)に関してはアクタの持つ属性の変化であると考へ、その変化後の値を示す。属性に関しては次の節で詳しく述べる。

d)のアクタ間の変化は、変化後の関係を記述する。

e)は対象とするアクタの動作のトリガと考へる。対象アクタの動作は上のa)~d)のどれかである。表現は対象アクタと共に動作を矢印で示す。

いくつかの動詞の格情報と、その用例、及びその動作の概念表現の例を以下に示す。格のタイプとして表1. に示すものを用いている[2,3]。動作の内容が時間的に不完全なアスペクトを指す場合はその前にs(開始)、e(終了)、d(最中)を付けて表している。

上がる1(Sh (Of) Od)  
 [Shが(Ofから)Odへ上がる]  
 MOVE.UP(Sh, Of, Od) & be\_on(Sh, Od)

上がる2(S Op)  
 [SがOpを上がる]  
 MOVE.UP(S) with PATH(Op)

上がる3(So (Op))  
 [Soが(Op(を/に))上がる]  
 MOVE.UP(So) in SPACE(Op)

開ける1(Sh O)  
 [ShがOを開ける]  
 O←“開く”

開ける2(Sh Oself)  
 [ShがOselfを開ける]  
 Sh.Oself←“開く”

降りる1(Sh (Of) Od)  
 [Shが(Ofから)Odへ降りる]  
 MOVE.DOWN(Sh, Of, Od)  
 降りる2(Sh Op)  
 [ShがOp(を/に)降りる]  
 MOVE.DOWN(S) with PATH  
 止まる(S (Od))  
 [Sが(Od(で/に))止まる]  
 e:MOVE(S) & be\_at(S, Od)  
 持つ(Sh O)  
 [SがOを持つ]  
 hold(Sh, O)  
 寝る(S (Op))  
 [Sが(Op)に寝る]  
 body(S) = LIE & be\_on(S, Op)  
 歩く(Sh (Op) (Of) (Od))  
 [Shが(Opを)(Ofから)(Odへ)歩く]  
 MOVE(Sh, Of, Od) in SPACE(Op)

格	意味	アクタ属性の制約
S	主体アクタ	
Sh	HUMAN型	
So	NON-HUMAN型	
Op	場所	・アクタの前/後/左/右 ・アクタ(shapeCARPET) ・アクタ(insideEXIST) ・O(sizeL)の上 ・領域
Of	場所の始点	同上
Od	場所の終点	同上
Oself	主体の体の一部	Sh.parts
O	動作対象アクタ	
Ow	道具	
Om	動作の相手	

表1. 格の種類

## 6. アクタとその動作の表現

ここではアクタと、その動作及びアクタ間の関係の表現について述べる。

アクタは各種の属性を持つ。属性によりアクタが行える動作を規定する。典型的な属性をクラス化したものから継承の機能を用いて、基本的な属性とその値を持たせることもできる。()は注釈、{}は値の候補またはその例を示す。

### ●アクタの持つ属性

- ・existence(存在)  
{T[場所]/F}
- ・type  
{人間的/非人間的}
- ・body  
{STAND/LIE}
- ・direction(方向性)  
{前後/左右/上下}
- ・parts(人ならば手、足、頭、胴体、目、口  
タンスならば本体とドア、など)

- ・size  
{LL(人より大きい)/L(かかえられる)/M(持てる)/S(つかめる)/SS(つまめる)}
- ・shape  
{ドア状/カーテン状/カーペット状/ロープ状/ボール状/コップ状/砂状/液状}
- ・inside(箱、たんす、バッグなど)  
{True/False}
- ・固有の属性(表示形態に対応することが多い)  
{OPEN/CLOSE}など  
({揺れている})のような継続/反復動作にある、というものを含む)

動作の指定と動作結果の表現のためアクタ間の関係として以下の種類を用意している。

### ●アクタ間の関係

- ・turn\_to(~を向いている)
- ・face\_to(至近距離にいる)
- ・be\_at(~のあたりにいる)
- ・be\_in(~の中にいる)
- ・be\_on(~の上にいる)
- ・aside(~の横にいる)
- ・hold(~を持っている)

実際の画面表示イメージに明確に対応しないものでもアニメーションシナリオの記述に必要な、領域や空間の指定を行うため、何らかの方法でそれを定義できるものとする。現在のところはいくつかのタイプを持つものとして名前参照する。

### ●領域の持つタイプ

- ・漠然とした空間型(SPACE)  
部屋、庭
- ・経路となる領域(PATH)  
階段、廊下
- ・接する面を指定するもの(SURFACE)  
壁際
- ・特定の位置(POINT)  
部屋の中央

これらを使ってアクタの動作を表現することが可能になる。

まず基本的に全てのアクタは移動が可能である。移動後の位置を、他のアクタとの関係を用いて表現する。ただし、「入る」「出る」「乗る」などは、その対象のアクタに内部(inside属性)が必要になる。

表示や向きに関する動作とは、direction属性を持つアクタのbody属性の値を変更することである。

形態の変化については、そのアクタが持つ、変化形状を指定する固有の属性の値を変更することで表現する。

以上の属性、関係を用いたアクタの表現を示す。

形式は以下の3つ組である。

アクタ表現:=

[アクタ名 [属性/関係リスト] [動作/属性対応]]

なおpartsについてはサブアクタとしてアクタ表現をネストさせる。動作/属性対応の部分にはそのアクタ特有の動作として記述される日本語動詞と、それによって変化させる属性及びその値を記述する。

例としてムーミンの表示形状をモデラなどを使って作成し、これをムーミンというアクタとして使う場合の表現例を示す。ムーミンは人間型のアクタであることから、デフォルトで前後左右などの方向性と、partsとして手、足、胴体、目、口などを持つ。どんなアクタでもアクタ自身の表示や向きに関する動作が可能である。

```
[ムーミン [[type HUMAN]
            [body +SIT1, +SIT2]
            [parts 目
                 [[figure OPEN/CLOSE]]
                 [[開く [figure OPEN]]
                 [閉じる [figure CLOSE]]]
            [すわる body(SIT1), body(SIT2)]]]
```

このほか一般にアクタは、固有の属性以外のデフォルトとして自身の存在位置や向きを表現する属性を持つ。

```
existence[True/False]
body[STAND/LIE]
turn_to[O]
face_to[O]
be_in[O]
be_at[O]
```

移動の例を示す。例えばベッドが以下のように表現されているとする。

```
[ベッド [[type NON-HUMAN]
          [direction ALL]
          [shape CARPET]
          [size LL]]
         [nil]]
```

ここで「上がる」という動詞の用法

上がる1(Sh (Of) Od)

を採用すれば、表1からOdの用件としてCARPET型のアクタがあるため、「ムーミンがベッドに上がる」の意味を

MOVE.UP(ムーミン, nil, ベッド)

と表現することができる。

次に箱、たんす、本、ボールなどその動作を行うための適当な対象物が存在すれば他動詞的な動作も可能である。例として本の定義を示す。

```
[本 [[type NON-HUMAN]
     [size MIDDLE]
```

```
[figure OPEN/CLOSE]]
```

```
[[開く figure(OPEN)]
```

```
[閉じる figure(CLOSE)]]]
```

これをもとにHUMAN型のアクタは「本を開く/閉じる」という動作が可能で、又sizeがM(中程度)ということから「本を持つ」「本を投げる」などの解釈も可能となる。

## 7. シナリオの表現

先に挙げた例文に合わせて、上の動作辞書をもとにした解析の例を示す。s/e/dは先と同じアスペクトの意味を表す。数字は文の番号であるとともに、動作の時刻でもある。

1. d:body(ムーミン) = LIE  
be\_on(ムーミン, ベッド)
2. ムーミン.目←“開く”
3. body(ムーミン) = STAND  
MOVE.DOWN(ムーミン, ベッド, nil)
4. s:MOVE(ムーミン, nil, たなの方)
- 5-. be\_on(本, 机の上)
5. be\_at(ムーミン, 本)  
hold(ムーミン, 本)
6. e:MOVE(ムーミン, nil, いすの前)
7. IN\_A\_WHILE/  
be\_on(ムーミン, いす)  
body(ムーミン) = SIT1|SIT2
8. 本←“開く”
9. MOVE(ノンノン, nil, 部屋)  
be\_in(ノンノン, 部屋)
- 10.d:MOVE(ノンノン, nil, nil) in 部屋
- 11.e:MOVE.FAST(ノンノン, nil, ドア)

この解析の過程で以下のようなデフォルト推論を用いている。[4]

- ・省かれた主語は変わらない。
- ・対象物が持っている関係を変更するような動作は、その変更する前の関係を“-”(マイナス)を付けた時刻で表現する。
- ・目標地点が明示されないMOVEはアスペクトsである。出発地点が明示されないMOVEはアスペクトeである。両方が明示されないものはアスペクトdである。
- ・HUMAN型のアクタのMOVEはbody(STAND)を前提条件とする。
- ・他動詞的な動作eに関しては、その動作対象物の前にいなければならない(be\_at)。かつ移動の最中ではない。
- ・移動の起点があれば、その移動が起こる瞬間にそこにいなければならない。

次に、これらの動作間の時間的な関係の表現について述べる。

ここで、日本語シナリオの解釈において、一つの文が一つの動作や状態を表現し、かつ記述の順序がそのまま時間の前後関係を表すと仮定している。

先に挙げた例文に合わせて時間軸に沿った動作の順序を図に示す(図5)

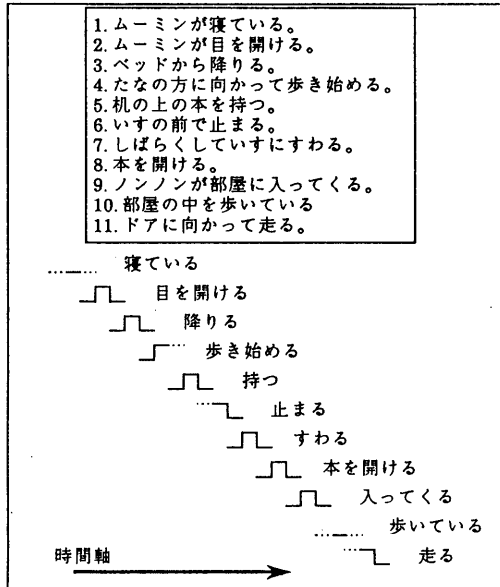


図5.

次にひとつひとつのアクタの属性、他のアクタとの関係、移動に関してまとめ、時間軸に沿って並べる(図6)。

時間を含む状態に関して、先のデフォルト推論に次のような規則を加える。

- ・開始や終了が明示されていない部分については、出来るかぎり長く状態を続ける。
- ・異なる種類のMOVEは片方を一旦終えて、次を始める

例えばムーミンのbody属性に関しては、以下ののような値の設定、及び変化がある。

1. d:body(ムーミン) = LIE
3. body(ムーミン) = STAND
7. body(ムーミン) = SIT1|SIT2

この場合は3.の直前まで“LIE”の状態が続き、3.から7.の直前まで“STAND”、あとは“SIT1”または“SIT2”の状態である、と解釈する。同じくムーミンの移動に関しては以下の記述がある。

3. MOVE.DOWN(ムーミン, ベッド, nil)
4. s:MOVE(ムーミン, nil, たなの方)

5. be\_at(ムーミン, 本)
- hold(ムーミン, 本)
6. e:MOVE(ムーミン, nil, いすの前)

3.と4.の移動の種類が異なるため、一旦3.のMOVE.DOWNは終了し、4.でまたMOVEを開始する。次に5でholdに関する動作を行っているため、ここで一旦移動をやめ、また6で移動の終了が述べられていることから5と6の間で移動を開始したことになる。すべてをまとめた結果を図7に示す。この結果をもとに、

- ・ムーミンはいすにすわる前に何をしたか
- ・本を持ってムーミンはどこからどこまで歩いたか
- ・ノンノンが部屋に入ってきたときムーミンは何をしていたか

などの質問に答えることができる。

### まとめ

アニメーション作成を目指した日本語シナリオの解析のモデル及びその解析処理について述べた。このモデルではアクタ自身とそれらの関係を属性で表現し、ある時刻における状態を登場するアクタとその属性の値の集合で表現した。またアニメーションにおける動作はその属性値を変更するものにとらえ、アニメーションシナリオを動作の系列として表現した。(図8)

今後の課題として、ゆっくりとした動きの表現や、時間を中心とした詳細な推論、常識の表現及びそれを使った推論について考察を進める。

### 謝辞

本研究は、通商産業省工業技術院大型プロジェクトの一環として(財)情報処理相互運用技術協会(INTAP)が新エネルギー産業技術総合開発機構(NEDO)からの委託を受け、シャープ(株)がINTAPからの再委託研究として実施したものである。

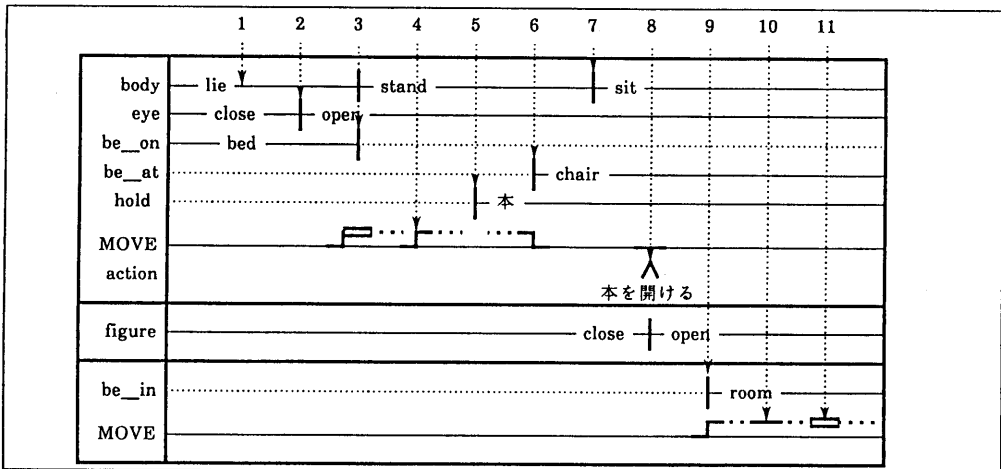


図6.

参考文献

[1]佐々木泰、岡田和久、高田正之、小谷善行:  
"知識ベースにおける動的概念の表現法",  
知識工学と人工知能 63-6(1989)、pp.49~58

[2]岡田直之、田町常夫:  
"動図形の意味解釈とその自然語記述 -意味分  
析-", 電子通信学会論文誌 Vol.J59-D  
No.5(1979)、pp.331~338

[3]岡田直之、田町常夫:  
"自然語および図形解釈のための単純事象概念の分  
析および分類", 電子通信学会論文誌 Vol.J56-D  
No.9(1973)、pp.523~530

[4]高島洋典、島津秀雄、友納正裕:  
"ストーリー駆動型アニメーション", NLC 86-  
18(1986)、pp.41~47

[5]小泉、船城、本田、仁田、塚本編:  
「日本語 基本動詞用法辞典」、大修館書店、1989

[6]森田良行:「基礎日本語辞典」、角川書店、1989

[7]宮本雅之、花田恵太郎、吉川耕平:  
"動画生成のための並行動作モデル", 情報処理学会  
第73回ソフトウェア工学研究会(平成2年7月)

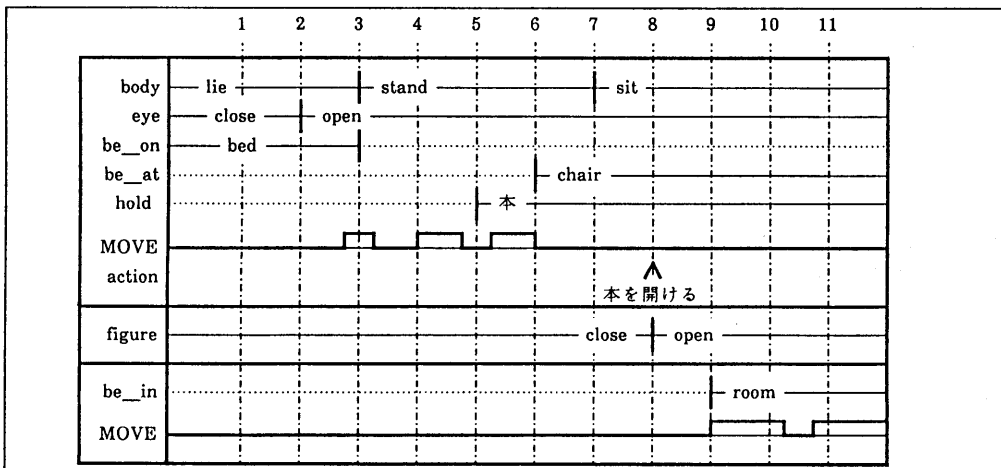


図7.