

Title	複数の動作を伴うアニメーテッドエージェントの行動を制御する対話システムの構築
Author(s)	藤澤, 瑞樹
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1446">http://hdl.handle.net/10119/1446</a>
Rights	
Description	Supervisor:奥村 学, 情報科学研究科, 修士

# 修士論文

## 複数の動作を伴うアニメेटドエージェントの行動を制御 する対話システムの構築

指導教官 奥村学 助教授

北陸先端科学技術大学院大学  
情報科学研究科情報処理学専攻

藤澤 瑞樹

2001年2月15日

## 要旨

近年、自然言語によるインタフェースの要望が高まって来ている。言語的インタフェースは、我々が普段から用いているため覚える必要がなく、複雑な指示を簡潔に伝えることが可能であるためである。また近年のコンピュータグラフィックス技術の進展により、機械的な制約を受けず、行動機能が豊富な3次元ソフトウェアロボットをコンピュータ内部に作り出し、それを自在に動かすことが出来るようになって来た。このような状況になり、新山らはユーザが仮想世界の映像を画面上で見ながら、仮想世界上のエージェントに対し音声を介して自然言語で指示を与えることが出来るシステム“傀儡”を開発した。傀儡は非常に有用なシステムであるが、仮想空間中のエージェントに対する令文に単文しか扱えないと言う問題を抱えている。我々は普段の生活で人に何か頼むとき、一回の発言で2個以上の動作を相手に頼むことが多い。すなわち複文を用いて頼み事をする場合が多いということである。したがって単文しか扱えないシステムでは、インタフェースとして不十分であると言える。

そこで本研究ではこの問題を解消し、傀儡を改良することでより良いインターフェースを持つアニメेटドエージェント行動制御システムを構築することを目的とする。

ここで複文を含む命令文を受け付けることが出来るようにすることは、一回の指示でエージェントに対し複数の動作を指示することを意味する。この複数動作の時間関係を特定し、それに即してアニメーションを生成しなければならない。本研究では、語彙概念構造に基づいた動詞のクラスを作成し、作成した動詞のクラスの組合せとそれに対応させる時間関係の規則を作成する。動詞クラスの組合せには、移動動詞を2つ組み合わせたもの、前項の動詞が後項の動詞の様態を表すもの、使役動詞を2つ組み合わせたもののうち前項の動詞が後項の動詞の手段を表すもの、使役動詞と移動動詞を組み合わせたもの、前項の動詞が主体の状態の変化を表すものなどを示し、それぞれがどのような時間関係を持つかを示す。また複文処理において、ある命令文ではその解釈が仮想世界の状況によって異なる場合がある。その場合の動詞クラスの組合せを示し、その際のアニメーション生成について示す。

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
1.1	本研究の背景	1
1.2	従来の言語理解システム	2
1.3	本研究の目的	3
1.4	本論文の構成	4
<b>2</b>	<b>傀儡について</b>	<b>5</b>
2.1	傀儡システムの概要	5
2.2	実行例	8
2.3	傀儡の問題点	9
<b>3</b>	<b>複数動作の制御</b>	<b>11</b>
3.1	複数動作制御の際の問題点	11
3.2	仮想世界	12
3.3	語彙的アスペクト	13
3.4	語彙概念構造	14
3.5	本研究の複数動作制御の手法	19
3.5.1	フレーム構造生成モジュールの改良	20
3.5.2	動詞のクラス	22
3.5.3	クラスの組合せとアニメーション	25
3.5.4	状況に依存した解釈	31
<b>4</b>	<b>まとめ</b>	<b>33</b>
4.1	結論	33
4.2	今後の課題	33



# 目次

2.1	傀儡の実行画面	6
2.2	傀儡のシステム構成	7
3.1	仮想世界全体	12
3.2	家, ロボット, ボール	12
3.3	静止状態の語彙概念構造	16
3.4	移動の語彙概念構造	16
3.5	状態変化の語彙概念構造	17
3.6	継続的活動の語彙概念構造	18
3.7	使役の語彙概念構造	19
3.8	改良後ののシステム構成	21
3.9	複文からのフレーム構造生成	23
3.10	必要でない格要素を移動させる	24
3.11	実行例 “ロボットは家の近くに寄って行って”	27
3.12	実行例 “ロボットはボールの近くに走って行って”	27
3.13	実行例 “ロボットはボールを投げて家を壊す”	29
3.14	実行例 “ロボットはボールを押して家の近くに行って”	30
3.15	実行例 “ロボットはボールを持って家に近付いて”	31
3.16	実行例 “ロボットはボールを持って” の後に, “ロボットはボールを持って家に近付いて”	32

# 表 目 次

3.1	Vendler の動詞 4 分類 . . . . .	13
3.2	英語動詞 4 分類のテスト基準 . . . . .	14
3.3	金田一の日本語動詞の 4 分類 . . . . .	15

# 第 1 章

## はじめに

### 1.1 本研究の背景

これまでの代表的なヒューマンインタフェースとしてマウスがあるが，最近マウスに現れるメニューの数が増え，単純な指令であってもマウスをクリックする回数が増えて来ている．また，近年の情報化の推進に伴って，計算機の使用に慣れていない人や高齢者，障害を有する人も計算機を使用する必要に迫られている．このような状況になり，近年，より自然で柔軟なインタフェースとして自然言語によるインタフェースの必要性が高まって来ている．

計算機のインタフェースには自然言語を用いたもの以外に，以下のようなものがある[田村 98]．それぞれの特徴とともに示す．

- コマンド言語

ユーザがシステムの操作に十分習熟すると，コマンド言語の使用により，効率の良い操作が期待できる．しかしコマンド言語は操作が分かりにくく，操作の習得に時間がかかる，操作エラーをしやすいなどの欠点を持つ．

- グラフィカルインタラクション (GI)

GUI(Graphical User Interface) はコマンドを記憶しておかなくても操作が可能であり，また情報伝達の際に図を用いると，瞬時にその内容を把握することが出来る．

- ノンバーバルインタラクション

ノンバーバルインタラクションを行うことによって，感性情報を直接伝えることが出来る．また明示的な表現を避けたり，同時に双方向の情報交換が可能となる．



- マルチモーダルインタラクション

音声言語やノンバーバルインタラクションなどさまざまなインタラクションの様式を機能させる。

これに対し自然言語によるインタフェースは以下のような特徴，利点を持つ。

- 人々にとってもっとも日常的なコミュニケーション形態である。
- 複雑な指示を簡潔に伝えることが可能である。
- 表現力が高く，どのような作業においても語彙が不足することはない。
- 複数の対象に指示を伝えたり，順序だった指示を効率的に与えることが出来る。
- その場の状況に依存して，指示の細かさを変えることが出来る。
- 前に与えた指示を修正したり，効率的に再利用することが出来る。

以上のような利点を持つため，自然言語を使用して制御できるシステムの登場が望まれる。しかしそのようなシステムは，今のところまだ実用化されていない。

## 1.2 従来の言語理解システム

計算機に自然言語を理解させるという研究は，Winograd の SHRDLU がよく知られている [Winograd76]。SHRDLU は積木の世界を扱っており，ユーザは計算機中のロボットに対し自然言語で指示し，積木を移動させることが出来る。そしてユーザはディスプレイ上に表示されたロボットの動作を観察することが出来る。しかし，この SHRDLU は積木の世界という非常に小さな世界を扱っていたためにその実現が可能であったといえる。

この当時から，自然言語でロボットの行動を制御する研究の重要性は指摘されていたにも関わらず，SHRDLU システム以後，研究の進展は必ずしもはかばかしくなかった。Winograd の成果を応用して，実際に実用的なシステムを作るには，依然として解決すべき問題が数多く残されているためである。それには以下のようなものがある。

- 機械的なロボットにしるコンピュータ内部の仮想空間のロボットにしる，当時はそれらの動作機能が限られていた
- 指令文を端末からいちいち入力する手間が問題であったこと

- 自然言語処理技術が未熟であったこと
- 計算機の性能が不十分であったこと

しかし近年，計算機の性能の向上，コンピュータグラフィックス (CG) 技術の向上により，計算機上に行動機能が豊富な 3 次元アニメテッドエージェントを作り出し，それを自由に動かすことが可能になってきた．このような状況になり，Badler らは，人間の身体の動きを正確に模倣できる 3 次元人体 CG システム Jack を開発した [Badler93]．Badler らはこの Jack システムを利用して，自然言語を使って人体モデルを制御する AnimNL プロジェクトを進めている [Badler99]．そして Badler らは昨年から言語と行動の視点を取り入れた研究に着手しようとしている．当面は副詞と動作との関係（「激しく動く」など）を扱うこととしており，本格的な言語理解の観点からの研究に着手する段階には至っていない．しかし，程度の副詞の理解などは，動作様態に関わるものであるので，特に重要である．

Stone らは CG アニメーションを用いた教育エージェントを学習環境に取り入れることによって，学習者がシステムにより関心を持つようにし，学習意欲を湧かせるようにした [Stone96]．

長尾は，これまでの GUI や直接操作とは異なり，より抽象度の高い間接的なヒューマンインタフェースとして，人間の顔を持ち，音声言語で対話するシステム Talkman を開発した [長尾 99]．

Strassmann は，自然言語によるインタフェースを使用して，既存の CG アニメーション生成ツールより簡単に操作できるようにする，Desktop Theater を開発した．Desktop Theater では，ユーザは脚本家となり，計算機上のキャラクタに自然言語により指示を与えることが可能である．そしてキャラクタはその自然言語を理解し，その指示に即してオフラインでのアニメーションを行う． [Strassmann94]．

新山らはユーザが仮想世界の映像を画面上で見ながら，仮想世界上のエージェントに対し音声を介して自然言語で指示を与えることが出来るシステム“傀儡”を開発した [新山 99]．傀儡では，音声認識システムの出力である単語の系列を文とみなし，それに対して構文解析や意味解析を行い意味理解の結果 (Semantic Representation/意味表現) を得る．この意味表現を更にロボットの動作指令に変換し，ロボットの動作映像を最終的に得ている．

### 1.3 本研究の目的

1.2 節で述べた傀儡システムは近年開発されたシステムの中では非常に有用なものであるが，傀儡はエージェントに対する命令文に単文しか扱えないと言う問題を抱えている．我々

は普段の生活で人に何か頼むとき，一回の発言で2個以上の動作を相手に頼むことが多い．すなわち複文を用いて頼み事をする場合が多いということである．したがって単文しか扱えないシステムでは，インタフェースとして不十分であるといえる．

そこで本研究ではこの問題を解消し，傀儡を改良することでより良いインターフェースを持つアニメテッドエージェント行動制御システムを構築することを目的とする．

複文を含む命令文を受け付けることが出来るようにすることは，一回の指示でエージェントに対し複数の動作を指示することを意味する．この複数動作の時間関係を特定し，それに即してアニメーションを生成しなければならない．本研究では，語彙概念構造に基づいた動詞のクラスを作成し，作成した動詞のクラスの組合せとそれに対応させる時間関係の規則を作成する．動詞クラスの組合せには，移動動詞を2つ組み合わせたもの，前項の動詞が後項の動詞の様態を表すもの，使役動詞を2つ組み合わせたものうち前項の動詞が後項の動詞の手段を表すもの，使役動詞と移動動詞を組み合わせたもの，前項の動詞が主体の状態の変化を表すものなどを示し，それぞれのアニメーションをどのように生成するかを示す．また複文処理の際，ある命令文では仮想世界の状態によってその解釈が変化する可能性がある．そのような場合の動詞クラスの組合せを示し，その際のアニメーション生成について示す．

## 1.4 本論文の構成

本論文はまず第2章で本研究の基となり，本システムを作成する際の土台となるシステム“傀儡”について述べる．第3章では命令文中に複数の動作が含まれている場合の処理の手法を提案し，その具体的なアルゴリズムを説明する．また作成したシステムについても簡単に解説する．4.1節では本研究のまとめを行い，4.2節で今後の課題を述べる．

## 第 2 章

# 傀儡について

### 2.1 傀儡システムの概要

新山らが開発した傀儡システム [新山 99] は、自然言語によるインタフェースを使用したシステムである。傀儡システムの実行画面を図 2.1 に示す。

ユーザはカメラを通して仮想世界の情報を得て、仮想世界上に存在するソフトウェアロボット<sup>1</sup> に対し音声による指示を与えることが出来る。ソフトウェアロボットはその指示を理解し、その命令に即して行動を取る。そしてユーザはカメラの映像でエージェントが命令を実行するのを確認することが出来る。このようにしてユーザはソフトウェアロボットと仮想的に共同作業を行うことが可能となる。

傀儡は大きく分けて以下の 3 つのモジュールで成り立っている。

1. 言語理解モジュール：与えられた命令文の構文解析，フレーム構造生成，意図抽出，照応解析，省略解析を行う。関数型言語 Scheme の一種である STk[Gallesio99] をプラットフォームとして利用している。
2. 音声認識モジュール：ユーザの音声を文字に直す。IBM の ViaVoice を使用して C++ で書かれている。
3. 3D グラフィックモジュール：仮想世界計算や視界計算，アニメーション生成などの 3D グラフィックスに関連する部分で，Carnegie Mellon University で開発された Alice99[CMU] を使用して，スクリプト言語 Python によって記述されている。

---

<sup>1</sup>仮想世界上のエージェントを本研究ではアニメータッドエージェントと呼んでいるが、新山らはソフトウェアロボットと呼んでいる。以下、ソフトウェアロボットとアニメータッドエージェントは同じ意味であるとする。

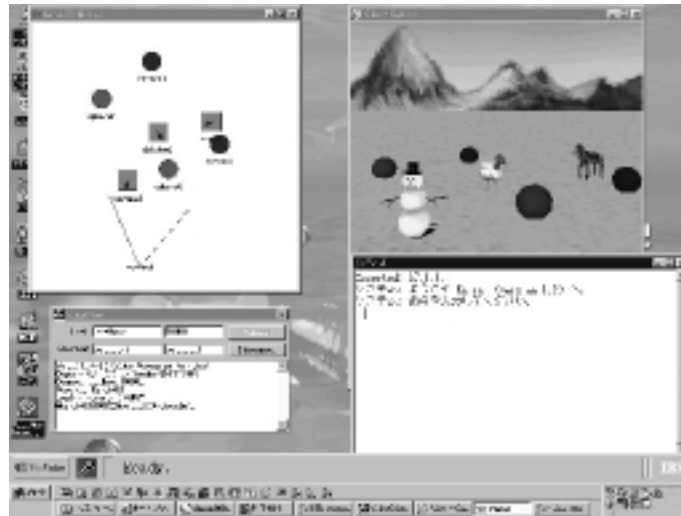


図 2.1: 傀儡の実行画面

傀儡のシステム構成を図 2.2 に示す .

傀儡の処理の流れは以下のようにになっている .

1. 音声認識モジュールによってユーザの発話した文を形態素列に変換し , これを構文解析モジュールに送る .
2. 構文解析モジュールは構文解析を行い , 構文木をフレーム構造生成モジュールに渡す .

フレーム構造は傀儡で用いられている中間表現である . フレーム構造は名前がついたスロットをいくつか持っており , 各々のスロットにはそれぞれ値あるいは別のフレーム構造を持つことができる . フレーム構造生成モジュールは入力された構文木を Filmore の格文法の理論に基づいてそれぞれの格に分け , フレームのスロットに入れる . 例として , 「ロボットはボールを持って」という文をフレーム構造で表現すると以下ようになる .

```
agent:   ロボット
object:  ボール
verb:    持つ
```

これは行為者 (agent) が「ロボット」で , 「持つ」という動作の対象物 (object) が「ボール」であることを表している . 動詞「持つ」は verb スロッ

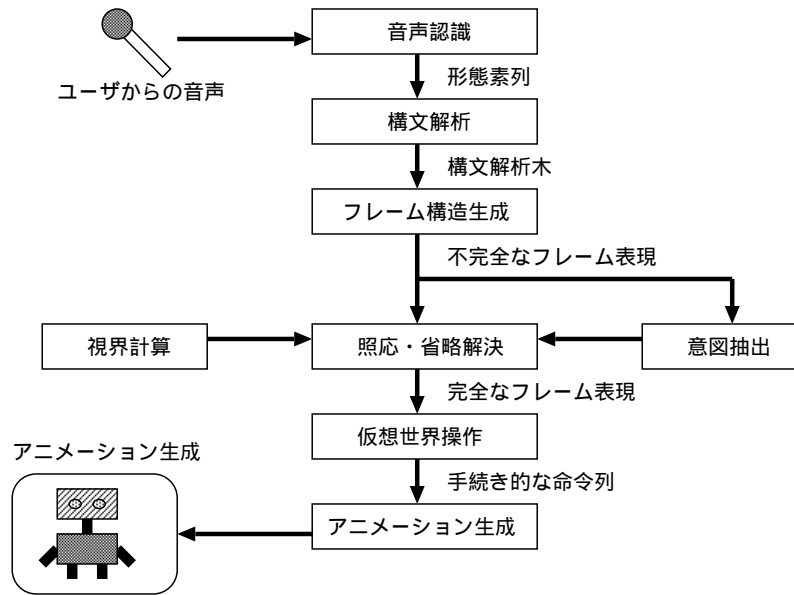


図 2.2: 傀儡のシステム構成

トに入れられる。

この時点で生成されたフレーム構造は、入力文に含まれる曖昧な表現や省略がまだ残っているため不完全なフレーム構造となっている。

3. 不完全なフレーム構造を、文中の名詞句が指す仮想世界内のオブジェクトが全て明確に決定された、完全なフレーム構造にするために、次に照応・省略解析モジュールに送る。

その際、不完全なフレーム構造はまず意図抽出モジュールに送られ、ユーザの意図を大まかに推測し、照応・省略解決の手がかりを得る。照応・省略解決モジュールはこれをもとに完全なフレーム構造を生成する。傀儡では、ロボットの目の前にある物体を「それ」などの代名詞で指示できるようにするために、照応・省略解決モジュールは仮想世界における各ロボットの視界情報も利用する。与えられた指示から対象が一意に定まらない場合は、システムからユーザに対し、質問を返す場合もある。

4. 完全なフレーム構造は仮想世界操作モジュールに送られる。仮想世界操作モジュールは仮想世界のデータベースを更新し、アニメーション生成のための手続きを生成する。

5. この手続きをアニメーション生成モジュールに送り，仮想世界が変化させられ，ユーザはその映像を観察する．

## 2.2 実行例

以下のような傀儡の対話例について考える．

傀儡の対話例 1．

1. ロボット A は赤い球をロボット B の近くまで押して
2. カメラはそれを写して

上記の命令文 1 が与えられると，まずロボット A は赤い球を探す．赤い球が複数ある場合は，ロボット A の視界情報をもとに押すべき球を決定する．そしてロボット B と探した赤い球を結ぶ直線の延長上に移動する．そして赤い球の方を向き，それをロボット B の手前まで押すことにより，命令文の実行を完了する．

次に命令文 2 が与えられると，カメラは“それ”を写そうとするが，字面の通りの“それ”では何を写せばよいのか分からない．そこでシステムは，照応はそれまでの文に出て来た名詞句のどれかを指し示すものととらえ「それ」が直前の命令文“ロボット A は赤い球をロボット B の近くまで押して”中の「赤い球」であるというように，照応の解決を行う．それにより，命令文 2 は“カメラは赤い球を写して”という命令文に置き換えられ，実行することが出来る．実行すると，カメラは赤い球を写すために移動を行い，ユーザは赤い球の写っている映像を観察することが出来る．

傀儡の対話例 2．

3. ロボットはその赤い球を押して
4. もう少し

上記の命令文 3 において，この文が対話の冒頭で話されており，これ以前に先行詞の候補となる名詞句が現れていないとする．この場合，命令文中に現れている「それ」は直示的な指示を表し，ロボットの前にある物体を指していると，システムはとらえる．命令文を実行するとロボットは前にある赤い球を適当な距離押してこの命令の実行を完了する．命令文 4 は文中の主語と述語が省略されている．命令文 4 は文単体では何が「もう少し」な

のか分からないが，その前の命令文<sup>3</sup>が“ロボットはその赤い球を押して”であることから，「(ロボットはその赤い球を)もう少し(押して)」という意味に解釈でき，命令文を実行することが出来る．

## 2.3 傀儡の問題点

傀儡を用いた自然言語理解の研究はまだ一昨年からはまったばかりであり，これから解決すべき研究課題はたくさん残っている [田中 00]．以下にそのいくつかを示す．

- 言語理解の機構に関する研究

- 音声認識と構文解析の統合化

音声認識システムは認識結果をただの文にして，それを構文解析器に送るが，音声には抑揚やイントネーションなどの情報も含まれているので，音声認識と構文解析を並列に行うとより良い結果が得られると考えられる．

- 意味理解機構

与えられた命令文を理解するために，その意図の抽出や，またエージェントの知識を知識ベースに蓄えておく必要がある．

- 言語生成

- 行動機能が豊富な3次元ソフトウェアロボットの構築

- 複数動作の実時間映像化

- 言語理解と身振りの関係解析

自然言語に画像，ジェスチャなどを統合しマルチモーダル化する必要がある．

- ソフトウェアロボットの表情の制御

- 言語と行動の統合に関する研究

- 動作のプランニングと学習

“家に入る”という命令文を処理するには，“ドアを開ける”という動作を実行すべきであると推論する必要がある．またその手順を知らない場合には，それを学習する必要がある．

- ソフトウェアロボットの協調動作



このうち本研究で扱う問題点は、傀儡は命令文中に複数の動作が含まれているときは処理できないと言うものである。これはすなわち傀儡はエージェントに対する命令文に単文しか扱えないということである。例えば、“ロボットはボールを家の近くに押して”のような命令文を処理することは出来るが、“ロボットはボールを押して家の近くに行って”のような動作を複数含む、複文を処理することは出来ない。しかし実世界において人は、このような複文を当り前のように使用し、対話を行っている。そのため、より自然で柔軟なインタフェースとして自然言語を用いようとするのならば、この複文の処理は是非とも解決しておくべき問題点であるといえる。そこで本研究ではこの複文の処理に焦点をしばって、その手法について述べる(3.5節) ことにする。

# 第 3 章

## 複数動作の制御

### 3.1 複数動作制御の際の問題点

複文を含む命令文を受け付けることが出来るようにするという事は、一回の指示でエージェントに対し複数の動作を指示することを意味する。この複数動作の時間関係を特定し、それに即してアニメーションを生成しなければならない。その際、複数動作の表現のうち、連続... ~してから... , ~する前に... , ~した後で... , .....

例： ボールを押す前に家の近くに行って

同時... ~する間... , ~しながら... , .....

例： ボールを押しながら家の近くに行って

のような動作間の時間関係を明示的に表している表現の処理は容易である。しかし、「~して...」は、命令文中の前項の動作と後項の動作を同時に実行するか、連続して実行するかが状況によって異なるため、これを処理するには工夫が必要となる。

複数の動作の時間関係をとらえるためには動詞のアスペクトを考慮しなければならない。そこで本研究では、アスペクト情報を取り入れている語彙概念構造を用いることとする。語彙概念構造を用いる利点として以下の点が挙げられる。

- アスペクト情報を扱うことが出来る。
- 各動詞ごとに規則をつけるのではなく、語彙概念構造をもとに分類したクラスごとに規則をつけるのでより一般的な方法で推論できる。

次節からはまず動作を複数含むような命令文を使用しやすくなるように変更した仮想世界について述べ、その後アスペクト、本研究で用いる語彙概念構造について述べる。そして次に本研究で提案する、複数の動作を含む命令文の処理手法を述べ、そしてその手法に必要なとされる語彙概念構造に基づいた動詞のクラス、クラスの組み合わせとその時間関係について述べる。

## 3.2 仮想世界

複文を用いた命令が出やすいようにグラフィックモジュールで用いている Alice99 を使用して、仮想世界を変更し、仮想世界中で用いる動作を追加した。作成した仮想世界は図 3.1, 図 3.2 のようになっており、これは仮想の「街」をイメージしたものである。仮想世界中にはドアのついている家や、木、自動車、ボール等が存在する。

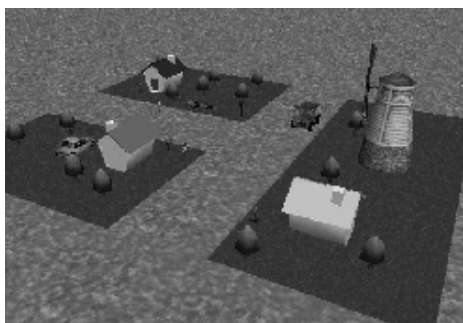


図 3.1: 仮想世界全体

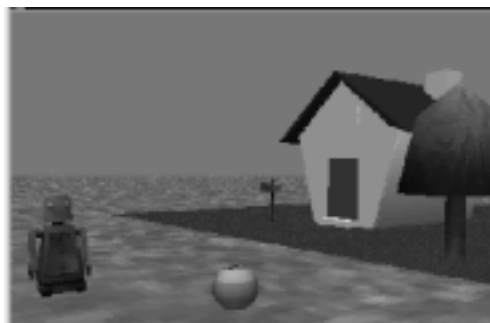


図 3.2: 家、ロボット、ボール

この仮想世界中で可能な動作は「行く」「歩く」「走る」「近づく」「寄る」「離れる」, 「遠ざかる」「持つ」「投げる」「開ける」「閉める」「置く」である。本来ならば、家に「入る」という動作が必要なのだが、家に入るという命令を受けた際、エージェントは以下のように実行を行わなければならない。

1. ドアの前に移動する。
2. ドアがしまっていれば、アのノブをつかみ、ノブを回してドアを開ける。
3. ドアを通過して家の中に移動する。

このように「入る」という動作を行うには、その命令から「ドアを開ける」などの動作を実行するべきであるということを推論しなければならない。しかしこのような推論機構は現在の傀儡は持っておらず、実行することは出来ない。この推論機構は今後の課題としたい(4.2節)。

### 3.3 語彙的アスペクト

完了, 未完了といった語彙的アスペクトが動詞の意味を考える上で重要であることは, 古くは古代ギリシアのアリストテレスから議論されて来た. ここでは現代の言語研究における動詞アスペクトの扱いの東西の代表として, 日本語に関する金田一と英語に関する Vendler という2つの研究について述べることにする [影山 96].

Vendler の研究は, 英語動詞を語彙的アスペクトによって分類したものとして言語学でも広く知られている. Vendler の動詞分類は表 3.1 のようになっている.

分類	例
(A) 状態 (states)	know, believe, have, desire, love
(B) 到達 (achievements)	recognize, spot, find, lose, reach, die
(C) 活動 (activities)	run, walk, swim, push a cart, drive a car
(D) 達成 (accomplishments)	paint a picture, make a chair, push a cart to the supermarket

表 3.1: Vendler の動詞 4 分類

直観的にいえば, 状態動詞 (A) は時間的な制限に縛られない恒常的な状態を意味する. 例えば know は何かについての知識が備わっていることを意味し, いつそれを知ったのか (開始) とか, いつ忘れてしまうのか (終了) といった時間的な境界は know 自体の意味とは無縁である. 本来的に変化のない状態を表すから, \*I am knowing him.<sup>1</sup> のように進行形をつける必要はない. もし状態動詞にわざわざ進行形をつけると, I'm believing his theory more and more. のように状態に至る過程を表すようになってしまう. (B) の到達動詞は, 何らかの目標 (状態) に至るという行為の終了点を重点的に述べる動詞であり, He is dying. のように進行形になると, その終了点に近付きつつあるという意味になる. 活動動詞 (C) は, 意図的に開始したり終了したり出来る行為を表し, 進行形をつけて He is walking. や He is pushing a cart. となると, その活動が目下, 継続中であることを意味する. 最後に, 達成動詞 (D) は何らかの活動の結果, 最終的な目標 (状態) に至ることを意味する. このことは, 単純な活動動詞と達成動詞を進行形に置いてみればはっきりする. He is pushing a cart. という活動は力の続く限り, いつまでも継続することが可能であるが, He is pushing a cart to the supermarket. という行為は目的地に到着することで達成され, そこで終了する.

---

<sup>1</sup>\*は非文を表す

Vendler はこのような違いを進行形の他にもさまざまな基準によって明らかにしている．代表的なテスト基準を表 3.2 に示す．

英語動詞 4 分類のテスト				
基準	状態	活動	達成	到達
進行形になる	no	yes	yes	?(意味が変わってしまう)
命令形になる	no	yes	yes	?
for an hour, spend an hour	Ok	OK	OK	bad
in an hour, take an hour to	bad	bad	OK	OK
for an hour, at all times in the hour	yes	yes	no	適切ではない
-ed in an hour, was	適切ではない	適切ではない	yes	no
		-ing during that hour		

表 3.2: 英語動詞 4 分類のテスト基準

日本語に関しては，Vendler に先んじて金田一が動詞の 4 分類を発表している．金田一は，状態，行為，変化といったアスペクトの観点に着目し，動詞に「～ている」がつくかどうか，また「～ている」がつく場合にはどのような意味になるのかを考察した．表 3.3 に金田一の日本語動詞の 4 分類を示す．

しかしこの金田一の分類も，先の Vendler の分類と同様に，基本的に動詞の分類に係わるだけで，意味構造そのものを述べたものではない．

### 3.4 語彙概念構造

動詞が表す概念的な意味を抽象的な述語概念で表示した構造を語彙概念構造 (Lexical Conceptual Structure, 略して LCS) または概念構造 (Conceptual Structure) という [影山 96], [郡司 98]．これは Jackendoff が提唱した考え方で，概念表現 (意味) のための基本要素の提案と，言語表現と概念表現との対応を説明するための理論である．この理論の背景にあるのは，全ての句範疇はある概念構造と対応し，語彙項目は引数が 0 以上の関数的な実体と

### 金田一の日本語動詞の4分類

第1種「状態動詞」: 時間の観念を超越して本来的に状態を表す動詞で、「ている」がつかない。[例] (机が) ある, (英語が) できる

第2種「継続動詞」: ある時間内続いて行われるような動作, 作用を表し、「ている」がつくと動作が進行中であることを意味する。[例] 読む, 書く, 笑う, 見る, 聞く, 置く, 歩く

第3種「瞬間動詞」: 瞬間に終わってしまうような動作, 作用を表し、「ている」がつくとその動作, 作用の結果の残存を意味する。[例] 結婚する, 卒業する, 死ぬ, 消える

「死んでいる」は死んでしまった状態, 「知っている」は知った後, 現在その知識を持っている状態を, それぞれ意味する。

第4種の動詞: いつも「~ている」の形で用いられ, ある状態を帯びていることを表す。[例] 優れる, 似る, ありふれる

表 3.3: 金田一の日本語動詞の4分類

対応するという仮定である。

#### a. 静止状態, 静止位置

まず Vendler 分類でも金田一分類でも基本的なグループである状態動詞から述べる。動きを伴わない静止状態は, さまざまなタイプの事象の中で最も基礎的なものであるといえる。静止状態ないし静止位置が基本的であることは, 概念構造にも反映されなければならない。何らかの物体がある場所に存在するというのがこの形式の基本的意味であるから, 仮に場所を  $z$  で, そこに存在する物体を  $y$  で表記すると図 3.3 のような概念構造が想定できる。

概念構造は, BE や AT のような述語とその項 ( $y, z$ ), およびそれらを統合した範疇 (STATE, LOC など) から成り立っている。図 3.3 は「 $y$  が  $z$  という位置にある」という意味を示し, 構造全体は状態 (STATE) という範疇で表されている。概念構造は図 3.3 のように枝分かれ図だけでなく, 全く同様に, 以下のような角括弧を用いて直線的に表示することも出来る。

$[_{STATE} [_{BE} [y, [_{LOC} AT z]]]]$  または簡略化して<sup>2</sup>  $BE[y, AT-z]$

<sup>2</sup> 今後は主に簡略化した形で示すことにする

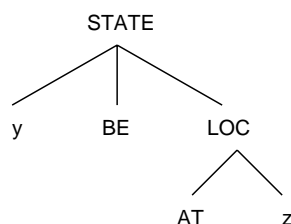


図 3.3: 静止状態の語彙概念構造

後者の表示法では，STATE などの標識を省略することもある．y や z は変項 (variable) と呼ばれ，統語構造では名詞句 (項) に対応する．

述語の BE は，さまざまな状態動詞に共通の「状態」という意味，すなわち特定の開始時間や終了時間を超越した継続性を表している．状態動詞が進行形にならないのは，この BE そのものが既に継続性の意味を固有に持っているからである．AT は位置を抽象化した概念で，日本語では「に」に対応する．

また，図 3.3 は物理的な位置だけでなく，抽象的な状態も “BE AT-z” という場所的な意味概念で表すことが出来る．以下の例文に置いて例文 1-1 は物理的な位置関係を表すのに対して，例文 1-2 は抽象的な状態を意味している．

例文 1: 1. 彼女は家にいる． ... BE[彼女 AT-家]

2. 彼女は健康だ． ... BE[彼女 AT-健康]

## b. 物理的な移動

物理的移動をを表す概念としては，GO を用いる．図 3.4 に GO の概念構造を示す．

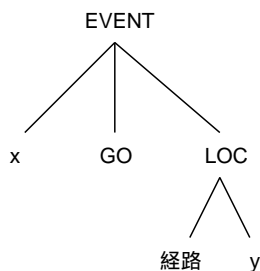


図 3.4: 移動の語彙概念構造

図 3.4 中の“経路”には，その移動の経路を表す概念が入る．経路を表す概念には，方向を表す TOWARD や位置関係を表す TO がある．また GO には修飾語句 (Adjunct) として，MANNER という様態を表す意味概念を持つことが出来る．様態とは，移動と直接係わる付随的概念で，様態を表す動詞には「転がる」や「流れる」などがある．

以下に GO を用いた例を示す．

例文 2： 1. 彼は家に行く． ... GO[彼 TO-家]

### c. 状態変化，位置変化

状態変化，位置変化を表す概念としては，BECOME を用いる．BECOME は Vendler 分類の到達動詞に相当する．また BECOME は二項述語で，第一項には主体を表す変項を取り，第二項には変化後の状態，位置を表す，前述の BE を用いた構造を取る．図 3.5 に BECOME の概念構造を示す．

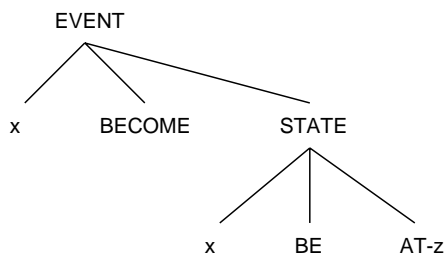


図 3.5: 状態変化の語彙概念構造

図 3.5 を直線的な表現にすると以下のようなになる．

BECOME[x, BE[x, AT-z]]

以下に BECOME を用いた例を示す．例文 3-1 は状態変化を表しており，例文 3-2 は位置変化を表している．

例文 3： 1. 箱が壊れる． ... BECOME[箱, BE[箱, AT-壊れている]]

2. シールが机に付く ... BECOME[シール, BE[シール, AT-机]]



#### d. 継続的活動

活動動詞に共通する意味特徴を表す概念としては ACT を用いる。ACT は Vendler 分類の活動動詞に相当する。また ACT は自動詞（一項述語）にも他動詞（二項述語）にも使え、二項述語の場合は ACT ON として表す。図 3.6 に ACT, ACT ON の概念構造を示す。



図 3.6: 継続的活動の語彙概念構造

図 3.6 を直線的に表示すると、

[EVENT [ACT[x]]] または簡略化して ACT[x]  
[EVENT [ACT-ON[x, y]]] または簡略化して ACT-ON[x, y]

また一項述語としての ACT に該当するのは「笑う」、「遊ぶ」などで、二項述語としての ACT ON に該当するのは「触る」、「たたく」などである。

ACT, ACT-ON は修飾語句として EFFECT という付帯変化を表す意味概念を持つことが出来る。付帯変化とは動作に伴って生じる状態変化のことで、付帯変化を持つ活動動詞には「立つ」や「投げる」がある。例として、「x は y を z に投げる」の概念構造は以下のようになる。

ACT-ON[x, y [EFFECT [GO[y, TO[z]]]]]

以下に ACT, ACT-ON を用いた例を示す。例文 4-1 は一項述語 ACT の例で、例文 4-2 は二項述語 ACT-ON の例、例文 4-3 は付帯変化を伴う例である。

- 例文 4: 1. 太郎が泣く。... ACT[太郎]  
2. 花子が箱を触る ... ACT-ON[花子, 箱]  
3. 次郎がボールを太郎のところに投げる ... ACT-ON[次郎, ボール, EFFECT[GO[ボール, TO-太郎]]]

## e. 使役

使役は、主語の何らかの行為によって、目的語の状態変化ないし位置変化が引き起こされることを意味し、それを表す概念としては CAUSE を用いる。CAUSE は Vendler 分類の達成動詞に相当する。また CAUSE は二項述語で、第一項には主体もしくは ACT(-ON) を用いた活動を表す構造を取り、第二項には GO を用いた移動を表す構造、もしくは BECOME をもちいた状態変化を表す構造を取る。図 3.7 に CAUSE の概念構造を示す。

$$\text{CAUSE} \left[ \left\{ \begin{array}{l} x \\ \text{ACT}(-\text{ON}) [x, (y)] \end{array} \right\}, \left\{ \begin{array}{l} \text{BECOME} [y, \text{BE} [y, \text{AT}-z]] \\ \text{GO} [y, \text{経路} - z] \end{array} \right\} \right]$$

図 3.7: 使役の語彙概念構造

以下に CAUSE を用いた例を示す。例文 5-1 は第二項が移動を表すもので、例文 5-2 は第二項が状態変化を表す例である。

- 例文 5: 1. 太郎は箱を頭の上に上げる ... CAUSE[太郎, GO[箱, TOWARD-頭の上]]  
2. 花子はボールを机の上に置く ... CAUSE[花子, BECOME[ボール, BE[ボール, ON-机の上]]]

## 3.5 本研究の複数動作制御の手法

傀儡では、単文の命令文しか扱っていないため、命令文中には一つの動作しか含まれていない。そのため一つの命令文からは一つのフレームしか生成されていない。しかし本研究では複文を用いた命令文の処理を目指しているため、命令文中には複数の動作が含まれることになる。そのため、複文を含む命令文からは、そこに含まれる各動作に対するフレームを生成することになるので、複数のフレームを生成しなければならない。そこで本研究ではまず、複文から、複数のフレームを持つフレーム列を生成するよう、言語理解モジュール中のフレーム生成モジュールを改良する。

これにより、複文を含む命令文からフレーム列を生成することが可能になるわけだが、実際に複文中の各動作を実行するには各フレームが持つ動作間の時間関係を特定しなければならない。この時間関係の特定には、動詞のアスペクト情報を考慮しなければならないため、本研究では 3.4 節で述べた語彙概念構造を用いることにする。この語彙概念構造を用

いて、動詞をいくつかのクラスに分けることにより、各動詞のアスペクト情報を利用することが可能になる。

そしてそのクラスを基に、どのクラスの動詞を組み合わせたときに、どのような時間関係を持つのかをあらかじめ規則化しておく必要がある。この規則を利用した、実際に複文を含む命令文の処理は以下ようになる。

1. 音声認識モジュールによって命令文を形態素列へ。
2. 構文解析モジュールによって構文解析を行う。
3. 改良を施したフレーム構造生成モジュールによって、複文を含む命令文の構文木を、各動詞ごとにフレームに分け、フレーム列を生成する。
4. 照応・省略解決モジュールによって、フレーム列中のフレーム全てを完全なフレームにする。
5. フレーム列中の各フレームに含まれる動詞のクラスを特定し、時間関係解決モジュールによってフレーム間の時間関係を特定する。時間関係特定モジュールはクラスの組み合わせとその際の動作間の時間関係に関する規則を持っている。
6. 仮想世界操作モジュールによって、特定した時間関係に即した、アニメーション生成のための手続きを生成する。
7. 手続きをアニメーション生成モジュールに送る。

またこのように改良した後の傀儡のシステム構成を図 3.8 に示す。点線で囲まれた部分が本研究での改良点である。

次節からは、まず複文からフレーム列を生成できるようにする手法について述べる。次に本研究で作成した、語彙概念構造に基づいた動詞のクラスを示す。そして最後にどのクラスを組み合わせたときにどのような時間関係を持つのかを表した規則を作成し、それを示す。

### 3.5.1 フレーム構造生成モジュールの改良

本研究では、傀儡のフレーム構造生成モジュールを改良し、複文の命令文からは、複数のフレームを含むフレーム列を生成できるようにする。以下にその例を示す。

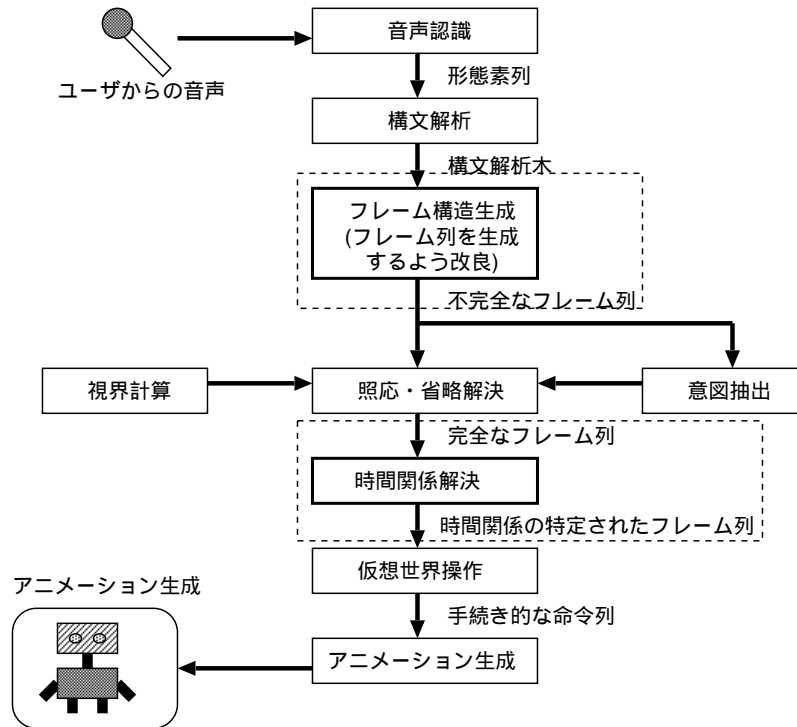


図 3.8: 改良後ののシステム構成

対話例：ロボットは家に近付いてボールを押して

この対話例は2つの動作を含んでいる複文の命令文である．そのためこの命令文からは2つのフレームが生成されなければならない．そのフレームは以下ようになる．Frame1は動作「近付く」に対するフレームであり，Frame2は動作「押す」に対するフレームである．

Frame 1.	Frame 2.
agent: ロボット	agent: ロボット
verb: 近付く	verb: 押す
object: 家	object: ボール

本研究での複文からのフレーム構造生成の手順を図 3.9 に示す．構文木を入力として受け取ると，まずフレーム構造生成モジュールは構文木から格要素を前から順に一つ取り出しそれをフレームのスロットにいれようとする．その際，もしそのフレームの verb スロット

が埋まっている場合は、現在のフレームのスロットに格納されている格要素のうち、verb スロットに格納されている動詞に必要な無い格要素を別のフレームに移す。格要素を移す候補となるフレームは新しく作成したものか、verb スロットの埋まっていない「未完成フレーム」のどちらかである。必要でない格要素を移した後、構文木から得られた格要素を前述のどちらかのフレームのスロットに格納し、始めに戻り処理を繰り返す。

またフレームの verb スロットが埋まっていない場合で、すでに構文木から得られた格要素を格納するスロットが埋まっている場合は、動詞により近いものを残すようにする。すなわち、既に格納されている格要素は別のフレームに移し、構文木から得られた格要素をスロットに格納することにする。

上記の処理を繰り返し、構文木に格要素がなくなったら終了となる。

上記の処理の説明において、フレーム中の動詞に必要な無い格要素を別のフレームに移すと説明したが、これを行わないとフレーム構造生成モジュールは構文木を前から順に見て行くので、例えば“ロボットはボールを家に近くに行って投げて”という命令文の場合、「ボールを」が「投げて」に係っているにも関わらず「行く」動作のフレームに格納されてしまう。そのため上記のような処理を行う必要がある。この処理は、図 3.10 のようにある動詞の前に同じスロットに入るべき格要素が 2 つ以上ある場合や、その動詞には必要のない格要素が動詞の前にある場合、それを後ろの動詞の方へ移動させて、構文木の変換を行っていると考えることが出来る。

### 3.5.2 動詞のクラス

本研究で作成した、語彙概念構造の構造に基づいた動詞のクラスを以下に示す。動詞のクラスは大きく分けて 6 つあり、それぞれのクラスはいくつかのサブクラスを持っている。

#### 1. 移動動詞 (第 1 述語として GO を持つもの)

##### 11. GO の第 2 項に方向を指定する述語をとるもの

111. 直示的方向を表すもの<sup>3</sup> 「行く、来る」

112. 非直示的方向を表すもの 「上がる、下がる、近づく、寄る」他

##### 12. GO の第 2 項に経路位置関係を指定するもの

「越える、渡る、通る、入る、出る」他

---

<sup>3</sup>直示的方向性を表す動詞は「行く」、「来る」の 2 つだけであり、それ以外の方向性を表す動詞を、非直示的方向性を表すものとする。

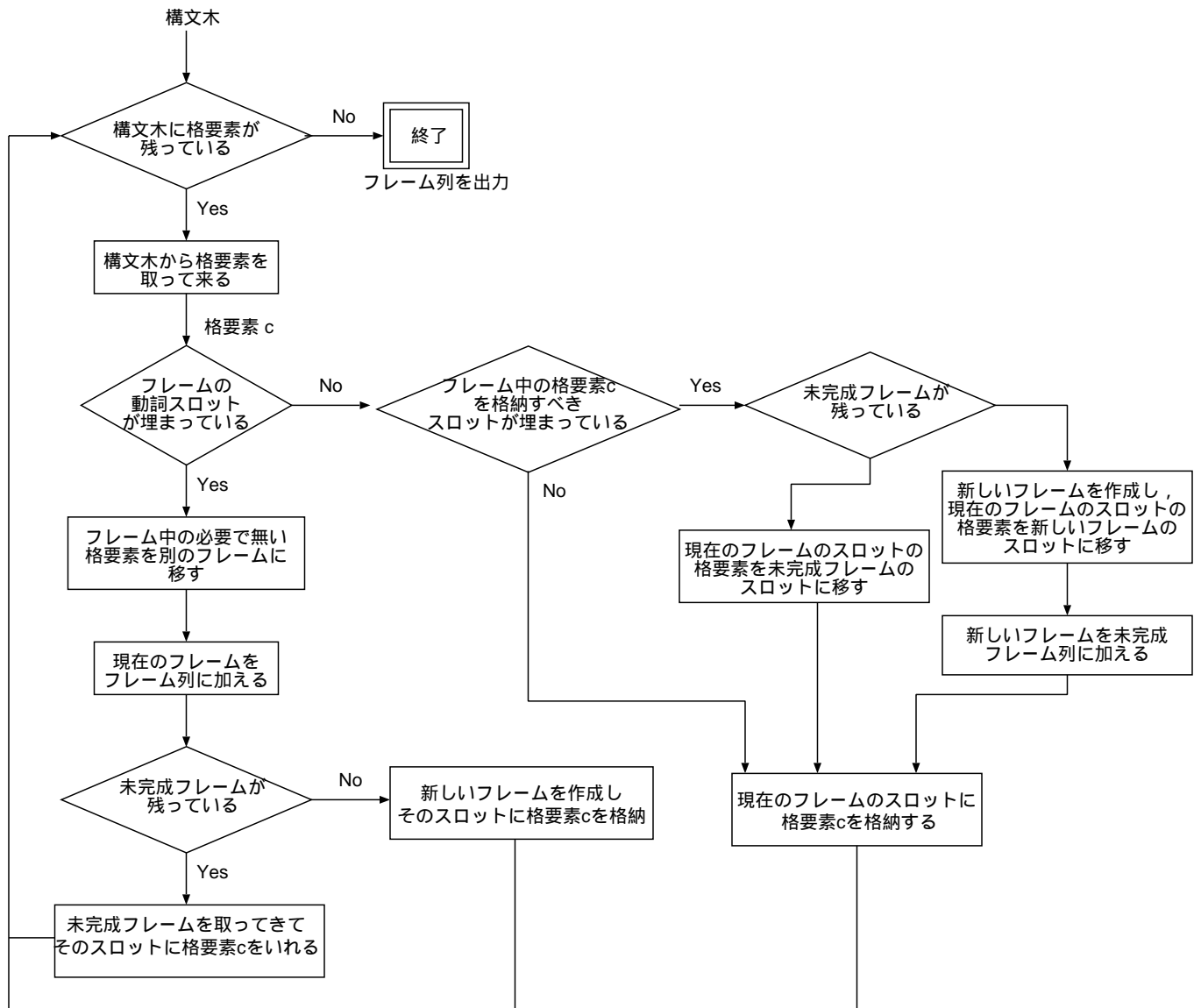


図 3.9: 複文からのフレーム構造生成

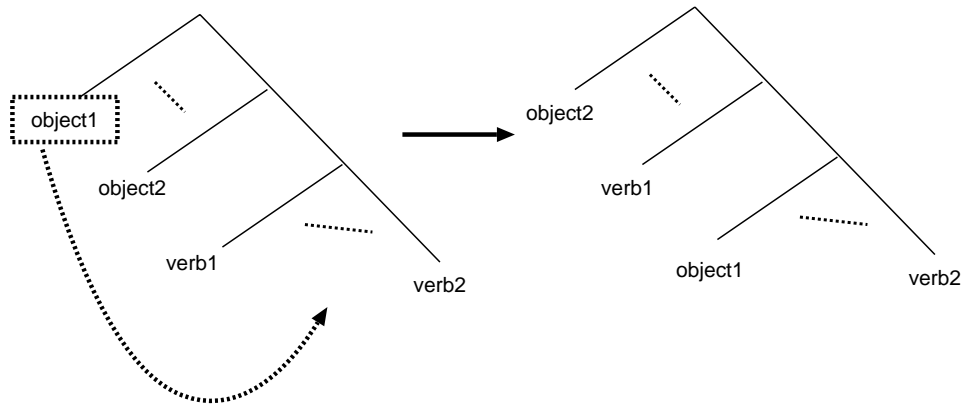


図 3.10: 必要でない格要素を移動させる

13. GO の Ajunct に MANNER を指定するもの  
「転がる、流れる、滑る」
2. 動作動詞 (第 1 述語に ACT を持つもの)
  21. 付帯変化として移動を伴うもの (ACT の Ajunct として、EFFECT という 1 項述語をとり、その項として GO を指定する)  
「歩く、走る、駆ける、泳ぐ、飛ぶ」他
  22. 付帯変化として主体の状態が変化するもの  
「立つ、すわる、寝る」
  23. その他  
「遊ぶ、泣く、笑う」他
3. 状態変化動詞 (第 1 述語として BECOME を持つもの)
  31. 移動に伴う状態変化 (BECOME の第 2 項に BE を、BE の第 2 項の状態として、移動の結果状態を位置を表す AT で指定する。前提条件として、分離の場合は付着していること、付着の場合は分離していること)
    311. 分離 「取れる、抜ける、はがれる、ちぎれる」
    312. 付着 「付く、つながる、刺さる、はまる」
  32. その他の状態変化 (BECOME の第 2 項に BE を、BE の第 2 項に状態を直接指定する)

「溶ける、色づく、壊れる、折れる、切れる」

4. 使役移動動詞 (第1述語として CAUSE を持ち、CAUSE の第2項に GO をとる)

- 41. CAUSE の第2項に 11 をとるもの  
「上げる、下げる、降ろす、落とす」他
- 42. CAUSE の第2項に 12 をとるもの  
「渡す、通す、入れる、出す」他
- 43. CAUSE の第2項に 13 をとるもの  
「飛ばす、転がす、流す、滑らす」他

5. 対象動作動詞 (第1述語として ACT-ON をとる)

- 51. ACT-ON の EFFECT として GO をとる  
「投げる、蹴る、押す、引く」
- 52. EFFECT を GO 以外または未指定  
「殴る、触る、たたく、打つ、にぎる、つかむ」  
(「持つ」はここに入る、ただし変化するのは行為の主体)

6. 対象状態変化動詞 (第1述語として CAUSE を持ち、CAUSE の第2項に BECOME をとる)

- 61. 移動の結果状態 (CAUSE の第2項に 31 をとるもの)
  - 611. 311 をとるもの 「取る、抜く、はがす、ちぎる」
  - 612. 312 をとるもの 「付ける、つなぐ、刺す、はめる」(「置く」はここに入る。ただし状態述語は ON)
- 62. その他の結果状態 (CAUSE の第2項に 32 をとるもの)  
「溶かす、壊す、折る、切る、開ける、閉める」

### 3.5.3 クラスの組合せとアニメーション

本節では 3.5.2 節で紹介した動詞のクラスに対し、その組み合わせがどのような時間関係を持つかを規則化する。以下にその規則を示す。



- {112, 12, 13} + {111, 21}, 動詞間に格要素を含まない

移動動詞を2つ組み合わせる場合, 下記の条件 [田中 97] を満たさなければならない。ただし “<” は “< の” 左項が右項より命令文中で前の位置にあることを表す。

付帯状況 < 様態 < 経路位置関係/否直示的方向性 < 直示的方向性

この条件を満たしているとき, 「移動動詞の表す移動は, その様態, 付帯状況と時間的に共起しなければならない」という時間的共起性の条件に従う。よってこの場合前項の動作と後項の動作を並列に実行する。ただし上記の条件中の付帯状況とは, 移動とは独立した付随的要素のことである。

対話例: ロボットは家の近くに寄って行って

この例文のフレーム列は以下のようになる。

Frame 1.	Frame 2.
agent: ロボット	agent: ロボット
verb: 寄る	verb: 行く
to: 家の近く	to: 家の近く

Frame 1 は複文中の前項の動作のもので, Frame 2 は後項の動作のものである。ここで前項の動詞「寄る」と後項の動詞「行く」のクラスは, 3.5.2 節から, 112 と 111 であり, かつ動詞は連続していて, 間に格要素を含んでいないので上記の条件を満たす。また「寄る」非直示的方向性を表す動詞で, 「行く」は直示的方向性を表す動詞なので, 語順の条件も満たして。よってこの複文中に含まれる動作「寄る」と「行く」は並列に実行する。図 3.11 にその様子を示す。

- {21} + {111, 112, 113}, 動詞間に格要素を含んでも良いが前と後項の to 格は一致していなければならない

前項が様態を表す動詞で後項が移動を表す動詞を組み合わせた場合, これも前述の時間共起性の条件に従う。よって前項と後項を並列に動作させる。

対話例: ロボットはボールの近くに走って行って

この例文のフレーム列は以下のようになる。

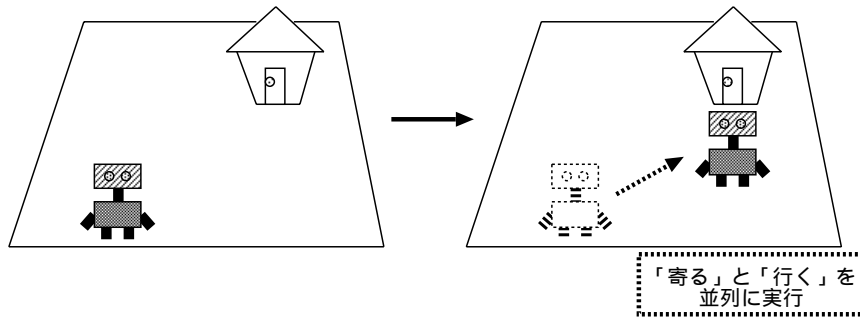


図 3.11: 実行例 “ロボットは家の近くに寄って行って”

Frame 1.	Frame 2.
agent: ロボット	agent: ロボット
verb: 走る	verb: 行く
to: ボールの近く	to: ボールの近く

Frame 1 は複文中の前項の動作のもので，Frame 2 は後項の動作のものである．ここで前項の動詞「走る」と後項の動詞「行く」のクラスは，3.5.2 節から，21 と 111 であるので上記の条件を満たす．よってこの複文中に含まれる動作「走る」と「行く」は並列に実行する．図 3.12 にその様子を示す．

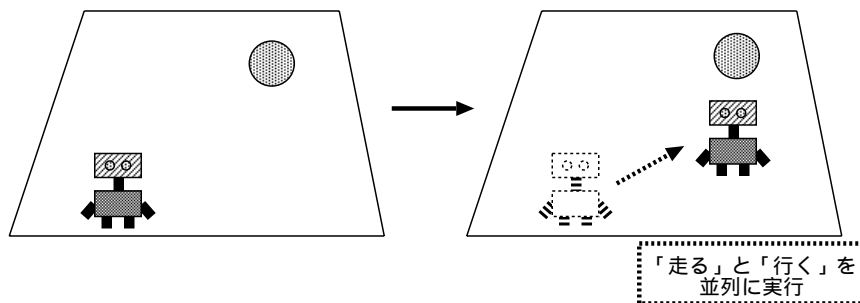


図 3.12: 実行例 “ロボットはボールの近くに走って行って”

- {41, 42, 43, 51, 52, 61, 62} + {41, 42, 43, 51, 52, 53}，動詞間に格要素を含んでも良い

使役動詞を 2 つ組み合わせる場合，使役手段を表すものが後項になることは無く，使役の手段を表す動詞は動詞の前項，経路位置関係/方向性/様態/付帯変化を表す使役

動詞が後項となる．このとき前項は後項の手段を表しているので，これは連続に実行する．ただし後項の動作は，その手段となる前項の動作の結果として生じるものなので，エージェントによって実行される他動詞的ものではなく，オブジェクトが自ら実行する自動詞的なものとする．またこの場合，前項の手段によって後項の動作が達成されないような場合は，その命令を受理することは出来ない．

このような前項の動作が手段となるかどうかの判断は，仮想世界に関する知識が必要となる．

本システムの仮想世界では，物を「投げる」ことによって「壊す」ことが出来，またボールを対象オブジェクトに投げつけることで対象オブジェクトを壊すことが出来るとして，前項が手段となる状況が出現するようにする．

対話例：ロボットはボールを投げて家を壊して

この例文のフレーム列は以下のようになる．

Frame 1.	Frame 2.
agent: ロボット	agent: ロボット
verb: 投げる	verb: 壊す
object: ボール	object: 家

Frame 1 は複文中の前項の動作のもので，Frame 2 は後項の動作のものである．ここで前項の動詞「投げる」と後項の動詞「壊す」のクラスは，3.5.2 節から，51 と 62 であるので上記の条件を満たす．よってこの複文中に含まれる動作「投げる」と「壊す」は連続に実行する．図 3.13 にその様子を示す．

- {51, 52} + {111, 112, 12, 13, 21}，動詞間に格要素を含んでも良い

移動動詞と使役動詞を組み合わせるとき，日本語では経路位置関係/方向性を主要部(後項)で表そうとするので，使役動詞を前項，移動動詞を後項にする．この場合，前項の動作が後項の動作の付帯状況となり，前述の時間的共起性の条件に従う．ただし前項の動作が瞬間的な場合，その繰り返しと移動が共起する(「投げて歩く」は何回も投げながら歩くを意味すると考えられる)．また前項の動作が繰り返し行えるものでないときは，命令中の動詞を受理することは出来ない．(“投げて歩く”の場合，対象となるオブジェクトを何回も投げることが出来るかで決まる)

対話例：ロボットはボールを押して家の近くに行って

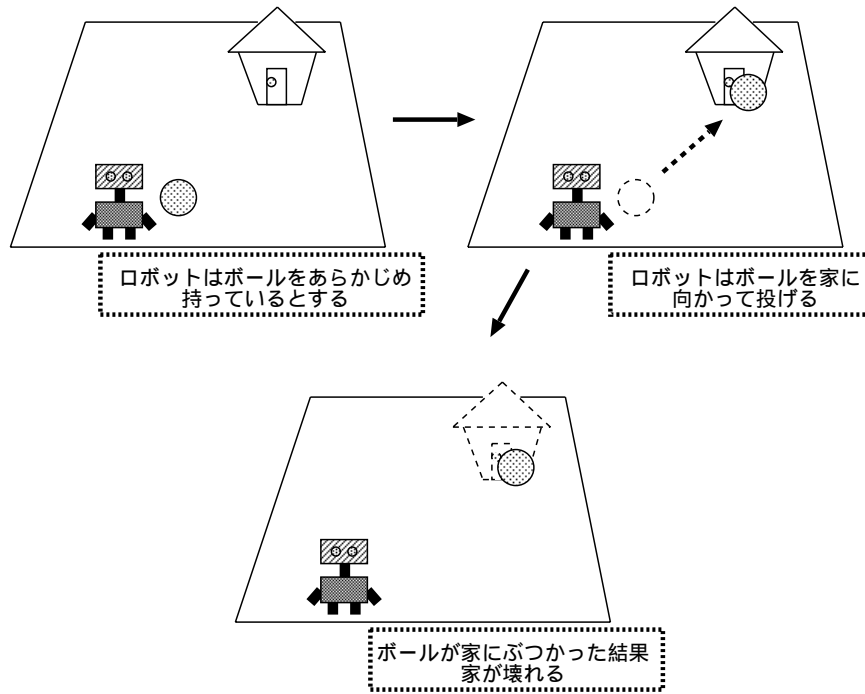


図 3.13: 実行例 “ロボットはボールを投げて家を壊す”

この例文のフレーム列は以下のようなになる。

Frame 1.	Frame 2.
agent: ロボット	agent: ロボット
verb: 押す	verb: 行く
object: ボール	object: 家の近く

Frame 1 は複文中の前項の動作のもので、Frame 2 は後項の動作のものである。ここで前項の動詞「押す」と後項の動詞「行く」のクラスは、3.5.2 節から、51 と 111 であるので上記の条件を満たす。よってこの複文中に含まれる動作「押す」と「行く」は、「押す」動作を繰り返し行いながら「行く」動作を実行する。図 3.14 にその様子を示す。

- {22, 52} + {全て}，動詞間に格要素を含んでも良い

前項が主体の状態変化を表している場合で、前項の動作は後項の動作の付帯状況となり、時間的に共起する。ただし動作の結果の状態が共起するので、動作は連続に実行させる。

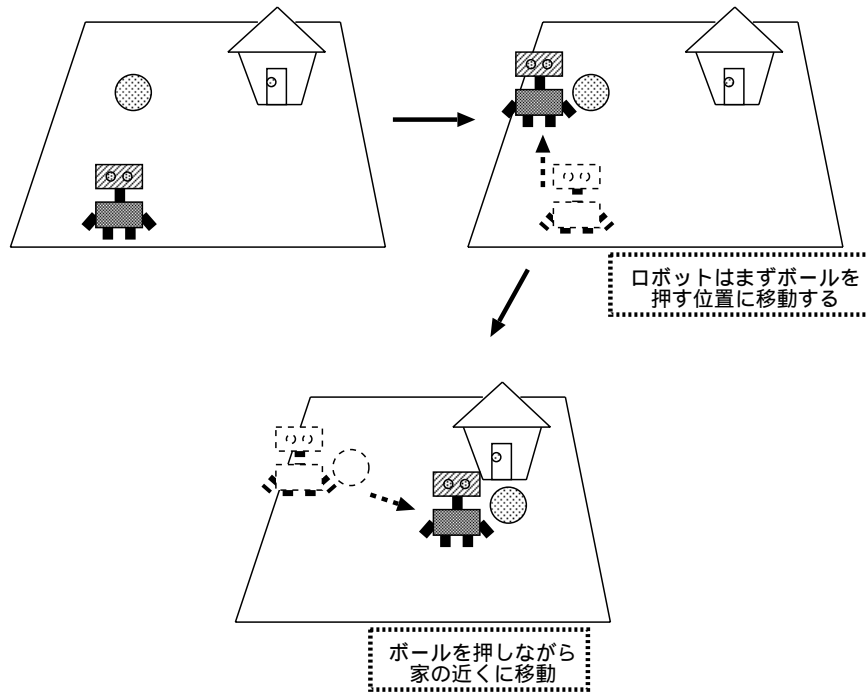


図 3.14: 実行例 “ロボットはボールを押して家の近くに行つて”

対話例：ロボットはボールを持って家に近付いて

この例文のフレーム列は以下ようになる。

Frame 1.

agent: ロボット

verb: 持つ

object: ボール

Frame 2.

agent: ロボット

verb: 近付く

object: 家

Frame 1 は複文中の前項の動作のもので，Frame 2 は後項の動作のものである．ここで前項の動詞「持つ」と後項の動詞「近付く」のクラスは，3.5.2 節から，52 と 112 であるので上記の条件を満たす．よってこの複文中に含まれる動作「持つ」と「近付く」を連続に実行する．ただし「持つ」によって生じる状態変化が保たれた状態で「近付く」を実行する．図 3.15 にその様子を示す．

- 上記以外の組合せにおいては，単に前項の動詞と後項の動詞を連続に実行させる．

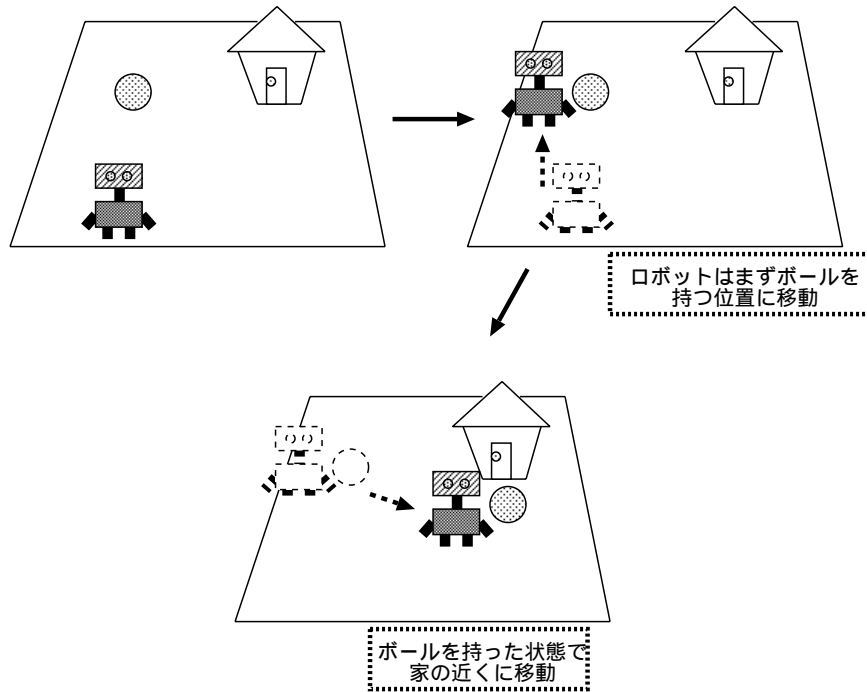


図 3.15: 実行例 “ロボットはボールを持って家に近付いて”

### 3.5.4 状況に依存した解釈

以下のような動詞クラスの組合せのとき，その命令の解釈は仮想世界の状況に依存し，異なる場合があると考えられる．

- {22, 41, 42, 43} + {全て}

命令文のパターンが上記のような場合で，前項の状態変化が既に満たされている場合，命令文の意図は前項の動作を実行するのではなく，その結果状態を持続したまま後項の動作を実行すると解釈することにする．

対話例： a. ロボットはボールを持って

b. ロボットはボールを持って家に近付いて

上記の対話例は，ユーザがまず a. の指示を行った後，b. の指示を行ったものとする．したがって b. の指示を行った時点では，ロボットはボールを既に持っている状態になっている．そのため，b. は通常は図 3.15 のように，まず「ボールを持つ」動作を行ってから「家に近づく」動作を行うのだが，この場合は既に「ボールを持つ」ことにより生じる状態変化後の状態になっているため，「ボール

を持つ」動作を行うのではなく、「ボールを持っている」状態で「家に近づく」動作を意図していると考えられる。

図 3.16 にその様子を示す。

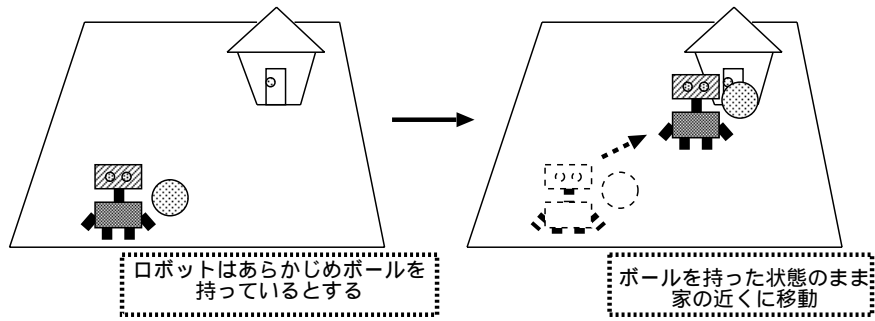


図 3.16: 実行例 “ロボットはボールを持って” の後に, “ロボットはボールを持って家に近付いて”

# 第 4 章

## まとめ

### 4.1 結論

本論文では、まず既存の言語理解システムを紹介した。それら言語理解システムの中で、特に新山らが開発したシステム「傀儡」を取り上げ、その問題点を指摘した。その内本研究で扱った問題点は、エージェントに対する命令に単文しか扱えないというものである。そこで本論文ではその問題を解消し、複文を含む命令文を受け付けることが出来るように改良するために、動作間の時間関係に必要なアスペクトについて紹介した。つぎにアスペクト情報を利用するために、本研究で用いることにした語彙概念構造について紹介した。次に語彙概念構造をの構造を基に動詞のクラスを作成し、そのクラスの組み合わせに対しどのような時間関係をとるかを規則化し、その規則を作成した。この規則を用いて時間関係解決モジュールを作成することにより、複文の処理が可能となった。また複文処理の際にある命令文のパターンではその解釈が変化する場合があることを示し、その場合の動詞クラスの組み合わせ、その際のアニメーション生成について示した。

### 4.2 今後の課題

本研究の今後の課題としては以下のようなものが考えられる。

#### a. プランニング機構の導入

動作を実行する際に、仮想世界の状態によっては不具合が生じる場合がある。例えば進行方向に障害物がある場合、それを避けるために経路を変更しなければならない。また家



に入るときには、ドアのノブをつかみ、ノブを回してドアを開ける動作が必要であることを命令文と仮想世界の状態から推論し、実行しなければならない。

本研究での複数動作の解釈においても、前項の動作を行ったために本当は後項の動作は実行することが不可能である場合などが生じる。また複数動作の時間関係の特定は、命令文からだけではなく、仮想世界の状態も考慮した上で行うべきである。

以上からシステム中にプランニング機構を組み込み、その動作が実行可能であるかどうかを調べ、不可能であるならば他の動作を選択できるようにすることが望ましい。

## b. 複数のエージェントに対しての指示

本研究での複数動作制御は単一エージェントを対象にしたものである。命令文の処理では的確に実行を行うことができるが、複数のエージェントに指示を行う場合の制御では不具合が生じる可能性がある。この処理には、上記で述べたプランニング機構が必要である。

また複数のエージェントに対して指示を行う際に、エージェント間で協調して動作を行うとより効率的に命令を実行できる。そのためこのような協調動作に関する研究も必要である。

## c. カメラの自動撮影

元の傀儡では世界が小さいため、カメラを移動しなくてもエージェントの行う動作をユーザは観測することができる。しかし今回用いる世界は傀儡で用いられていたものと比べて、世界が大きくなっているため、注目すべきエージェントやオブジェクトのところにカメラを移動しなければ、ユーザはその振舞を観測することができない。またここで元の傀儡のようにカメラの移動をユーザのみに任せてしまうと、ユーザに多大な労力を課すことになり、システムの使い勝手が非常に悪くなってしまうと考えられる。そこでエージェント(カメラ以外)に対するユーザの命令に応じて、カメラの移動を自動化する必要がある。

## d. 質疑応答が出来るようにする

現在のシステムでは、ユーザはカメラを通してしか仮想世界の状態を知ることが出来ない。仮想世界の十分な知識を得ることが難しい場合がでて来る可能性がある。そこでユーザがシステムに対し、仮想世界の状態を質問し、システムはその質問に返答を返すことが出来るようにする必要がある。

# 謝 辞

本研究を進めるにあたり，研究の機会を与えられ御指導いただいた奥村学助教授に心から感謝致します．また，御指導，御討論をいただいた島津明教授に深く感謝いたします．

本研究全般に渡りまして，熱心に御指導していただき多くの助言をしていただきました明星大学の石亨助教授に深く感謝致します．

適切な助言をして下さいました，望月学助手，自然言語処理学講座の皆様には感謝致します。

素晴らしいシステムを開発され，またシステムについて丁寧な後説明をして下さった東京工業大学徳永研究室の新山祐介さんに感謝します．

最後に，これまで私を育て，好きなことをさせて下さいました，両親に心から感謝致します．

皆様ありがとうございました．

## 参考文献

- [Badler93] Norman I. Badler, C. B. Phillips, and B. L. Webber, “Simulating Humans”, Oxford University Press, ISBN0-19-507359-2, 1993
- [Badler99] Norman I. Badler, Martha S. Palmer, and Rama Bindiganavale, “Animated Control for REAL-TIME VIRTUAL HUMANS”, Comm. of the ACM, Vol.42, No.8, pp.65-73, 1999
- [CMU] Carnegie Mellon University, “Free Easy, Interactive 3D Graphics for the WWW”, <http://www.alice.org>
- [Gallesio99] Erick Gallesio, “STk Reference Manual version 4.0.1”, Universite’de Nice - Sophia Antipolis Laboratoire I3S, 1999, <http://kaolin.unice.fr/STk>
- [Stone96] Brian A. Stone, James C. Lester, “Dynamically Sequencing an Animated Pedagogical Agent”, AAAI-96, Proceedings Volume One, pp.424-431, 1996
- [Strassmann94] Steve Strassmann, “Semi-Autonomous Animated Actors”, Proceedings of the 12th. National Conference on Artificial Intelligence, Vol.1, pp.128-134, 1994
- [Winograd76] Terry Winograd 著, 淵一博, 田村浩一郎, 白井良明訳, “言語理解の構造”, 産業図書, ISBN4-7828-5236-3, 1976
- [Zhao00] Liwei Zhao, Monica Consta, Norm I. Badler, “Interpreting Movement Manner”, Proc. Computer Animation 2000 Conference, IEEE Computer Society, Philadelphia, May 3-5, pp.112-120, 2000
- [影山 96] 影山太郎, “動詞意味論”, くろしお出版, ISBN4-87424-130-1, 1996
- [郡司 98] 郡司隆男, 阿部泰明, 白井賢一郎, 坂原茂, 松本裕治, “岩波講座 言語の科学 4 意味” 岩波書店, ISBN-00-010854-9, 1998

- [新山 99] 新山祐介, “ソフトウェアロボットの行動を制御する対話システムに関する研究”, 東京工業大学 情報理工学研究科 計算工学専攻 修士論文, 2000
- [田中 97] 田中茂範, 松本曜著, “空間と移動の表現”, 研究社出版, ISBN4-327-26006-1, 1997
- [田中 00] 田中穂積, “言語理解とロボットの行動制御 - 音声認識から音声理解へ -”, 信学技報 NLC2000-33, pp37-42, 2000
- [田村 98] 田村博, “ヒューマンインタフェース”, オーム社, ISBN4-274-07860-4, 1998
- [長尾 99] 長尾確, “エージェント指向インタフェース”, bit Vol.31 No.2, pp25-34, 1999