平成 14 年度

修士学位論文

ソフトウェアエージェントへの階層化複数決定木を用いた観察による学習の適用

Applying of Learning by Observations with Hierarchical Multiple Decision Trees to Software Agents

1055110　　平山　純一郎

指導教員　　竹田 史章

2003 年 1 月 31 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報システム工学コース

要 旨

ソフトウェアエージェントへの階層化複数決定木を用いた観察による学習の適用

平山　純一郎

ソフトウエアエージェントの適切な行動をハンドコーディングすることは非常に困難である．この問題を解決する方法の一つに強化学習があげられる．強化学習は，エージェント自身の経験により最適な行動が得られる可能性がある．しかし，複雑な動きを学習させるためには非常に膨大な時間が必要であり必ずしも適切な行動が得られない場合がある．そこで，人間がソフトウエアエージェントと同じ環境において行う意思決定挙動を基に学習する，観察による学習が提案された．本稿では，階層化複数決定木を用いて RoboCup エージェントへの観察による学習の適用する事を試み，その有効性について検証する．

キーワード　観察による学習，エージェント，RoboCup，決定木

Abstract

Applying of Learning by Observations with Hierarchical Multiple Decision Trees to Software Agents

Junichiro Hirayama

It is a difficult task to hand-code optimal actions for software agents. A solution to this is reinforcement learning. In reinforcement learning, agents might be able to acquire optimal actions by learning from their experiences. However, acquisition of complicated actions might take a great amount of learning time and might even be infeasible. To solve these drawbacks, an approach called learning by observations has been proposed in which learning of an agent is performed by observing human actions in the same environment of that of the software agent. In this paper, we discuss a novel learning methodology that uses hierarchical multiple decision trees, and apply it to RoboCup agent learning by observations proposed earlier by the authors. The effectiveness of the novel methodology is shown at the end of the paper.

key words Learning by Observations, Agents, RoboCup, Decision Trees

目次

図目次

表目次

第 1 章

はじめに

近年，ソフトウェアエージェント（以下エージェントとする）技術への需要が多くなってきている [1]．エージェントとは，コンピュータ上で自律的に動くソフトウェアのことである．エージェントの応用範囲として，情報検索，情報収集，ユーザインタフェイス，コミュニティ活動支援など広範囲に及んでいる．エージェントがこれらのタスクを効率よく行うためには，適切な行動，振る舞いをプログラムする必要がある．しかし，エージェントの適切な行動を定義することは，非常に困難な作業である．この作業を，改善するために以前から様々な手法が用いられてきた．本稿では，エージェントの行動を容易に定義するために，観察による学習を用いた．観察による学習を RoboCup サッカーシミュレーションリーグ [2] のエージェントに適用する方法と，その効果を検証した．

第 2 章

ソフトウェアエージェント行動定義

エージェントの行動を定義する方法として，大きく分けて 2 つあげられる．

- IF-THEN ルール等によるハンドコーディング
- 強化学習 [3] に代表される教師無し学習

ハンドコーディングによる行動の定義は，人間が考えられる状況に対してルールを定義することになる．単純なタスクを処理する場合は簡単に定義することができる．しかし，複雑なタスクの行動を定義する場合には，状況の数が多くなり人間では把握できなくなる．もし，状況がすべて把握できたとしても，それぞれの状況に対してルールを定義しなければならない．定義しなければならないルールは，複雑になり巨大な物になるそのうえ，人間が持っている知識やルールを実際に定義する事は非常に難しいことである．

一方，教師無し学習では，行動の定義をエージェントが自ら行動することにより得られる．単純なタスクに対しては，行動を定義することができる可能性がある．また，人間では見つけることが難しい最適解が見つけられる可能性がある．しかし，単純なタスクであっても人間が環境設定を適切なものに設定していないと，学習が収束せずに行動を定義する事ができない．環境設定を決めることは，人間の経験や能力が必要になり困難な作業である．また，タスクが複雑になると環境設定がさらに困難になり，適切な行動が得られないことや，適切な行動を得るまでに非常に多くの時間を必要としてしまう．

そこで，これらの問題を解決するために観察による学習が提案された [4]．これらエージェントの行動を定義する 3 つの方法の関係の概念図 2.1 に示す．

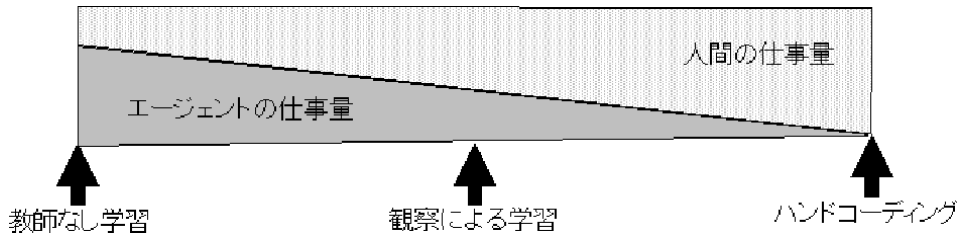

図 2.1　エージェント行動定義の位置関係

観察による学習は，エージェントの行動を定義するために，人間の意志決定挙動を基に行動の定義を行う．人間の意志決定挙動を得る方法としては，実際に人間がエージェントと同じ環境下にて状況判断を行い，その行動のログを取る．ログを基に定義された行動をエージェントに適用する．人間の仕事量は，教師無し学習よりも増えるが環境設定の仕事量が軽減され，行動が得られるまでの時間が短縮される．また，行動が複雑になってもハンドコーディングのように，複雑で巨大なルールを人間が定義する必要が無くなる．さらには，複雑な行動や他のエージェントとの協調行動，従来の手法では実現の難しい，人間の直感や経験の実現が可能であると考えられる．

第 3 章

観察による学習の RoboCup ソフトウェアエージェントへの適用

観察による学習を本稿では，RoboCup サッカーシミュレーション（以下 RoboCup シミュレーションとする）のエージェントに適用した [5].

また，観察による学習を適用するにあたり，観察による学習システムの構築を行った.

3.1 RoboCup について

RoboCup は, 人工知能と知的ロボットに関する研究を促進するためのものである. 自律移動ロボットによるサッカーを題材として, 日本の研究者たちによって 1995 年に提唱された. 現在では, サッカーだけではなく, 大規模災害シミュレーションや災害救助ロボットの研究・開発を行なう, RoboCup レスキューなどがある.

現在，RoboCup サッカーには，５つのリーグがある.

- シミュレーションリーグ

  コンピュータ上の仮想フィールド上で, 1 チーム 11 体のソフトウェアエージェント同士によって行なわれるリーグ. RoboCup サッカーの中では一番古くから存在するリーグで，一番洗練されたチームプレイをする． 本稿ではこのリーグを扱う.

- 小型ロボットリーグ

  卓球台とほぼ同じ大きさのフィールドで, 直径 18cm 以内に入る小さなロボットが 5 台以 内でチームを組み, オレンジ色のゴルフボールを使って対戦するリーグ.

3.1 RoboCup について

- 中型ロボットリーグ

  卓球台 9 枚分の大きさのフィールドで, 直径 50cm 以内に入るロボット 4 台でチームを組み, オレンジ色のフットサルのボールを使って対戦するリーグ.

- Sony 4脚ロボットリーグ

  SONY のエンターテインメントロボットによる 3 対 3 で行うリーグ. このリーグは，選抜されたチームに貸与されたロボットで競技を行うため，一般参加は出来ない.

- ヒューマノイドリーグ

  人間型二足歩行の自律ロボットによるリーグ. ROBOCUP サッカーの最終目標のリーグ.

毎年 RoboCup では, 国内大会と世界大会が行なわれる. ここで, 各チームの研究・開発されたロボット同士が勝負をしています.

RoboCup サッカーの最終目標は, ヒューマノイドリーグのチームが 2050 年に FIFA のワールドカップの優勝チームに勝つことである.

本稿で扱うシミュレーション部門は，SoccerServer と呼ばれるサーバ上で，UDP/IP で接続されたエージェント同士がサッカーの試合を行う，サーバ／クライアント方式を採用している．図 3.1 参照．エージェントは，それぞれ独立に動かなければならず，集中制御することは許されない．エージェントを作成する場合には, UDP/IP 通信をサポートするプログラム言語であれば何でも良い. 本稿では, Java を使用している.

サーバ上で行われるシミュレーションは，1 サイクル 100msec でおこなわれ，この単位サイクルに 1 つの行動することができる．実際のサッカーに近づけるために様々な制約がもうけられている.

1 つ目は，現実世界に近づけるために，物体 (エージェント，ボール) は予期せぬ動きをするように，ノイズと風が加えられる

2 つ目に，エージェントの行動は以下のものに限定される.

- turn　エージェントの向いている方向を体ごと変える.
- turn_neck　エージェントの向いている方向を頭だけ変える．体の方向は変わらない.

- dash　エージェントの体が向いている方向に加速をつける．
- kick　ボールが蹴れる範囲内 (kickable_area) にある場合に，方向と強さを決めてボールを蹴る．
- move　得点が決まったとき等の初期フォーメーションに戻るためと，ゴールキーパーエージェントが catch した時，ゴールキック位置に移動する時のみ使えるの行動
- catch　ゴールキーパーエージェントがボールをキャッチする行動．ゴールキーパーエージェント以外やペナルティーエリア外では無効．
- say　メッセージをすべてのエージェントに伝える．
- change_view　視野角度と視覚情報の品質を変える．これを変えると視覚情報の送られてくる頻度が変わる．

これらの行動を，1サイクルに1つ実行することが出来る．ただし，turn_neck,say,change_view は，他の行動と同じサイクルに実行することが出来る．また，エージェントにはスタミナのパラメータがあり，スタミナの残量によって行動が制限される．物体の動きと同様に，それぞれの行動にはぶれが生じる．

3つ目に，サーバから得られる情報に制約がある．サーバから得られる情報は以下のものである．

- 視覚情報　エージェントの視野内に入った物体 (エージェント，ボール) の距離，方向，物体の種類等の情報．
- 聴覚情報　審判からの判定メッセージと，エージェントが say コマンドで送ったメッセージ情報と聞こえた方向の情報．
- 感覚情報　自エージェントのスタミナ，視野モード，速度，頭の向き，kick, dash, turn, say, turn_neck のコマンドを送った回数の情報．

視覚情報と，聴覚情報のエージェントからのメッセージ情報は距離によって制限される．視覚情報では，近くにあるものは，距離，方向ともにかなり正確な情報が得られる．しかし，遠くにあるものは大まかな情報しか得られなくなる．聴覚情報のエージェントからのメッ

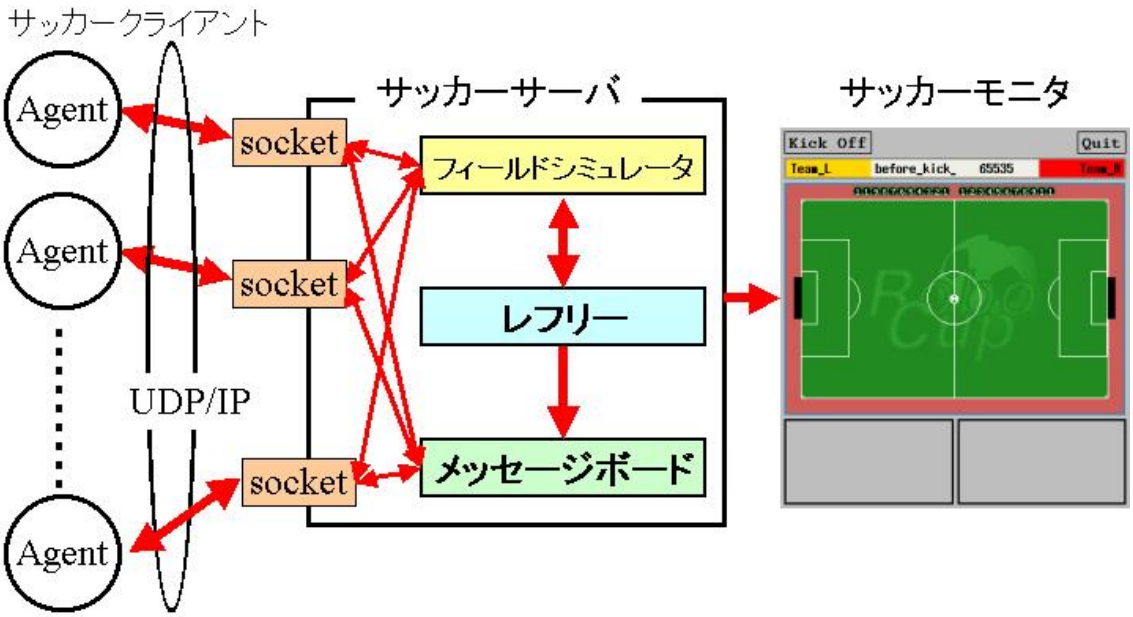

図 3.1　シミュレーションリーグ概要

セージはある一定距離離れると，聞き取ることが出来なくなる．

これらの制約を考慮して，エージェントを設計する必要がある．

3.2　観察による学習システムの構築

RoboCup エージェントへ観察による学習を適用するためのシステムを構築した．システムの構成図を 3.2 に示す．

この，システムは 3 つのシステムから構成されている．サーバから得られる情報は以下のものである．

- リアルプレイヤシステム
- ルール作成システム
- Huma システム

これらについて，以下で説明する．

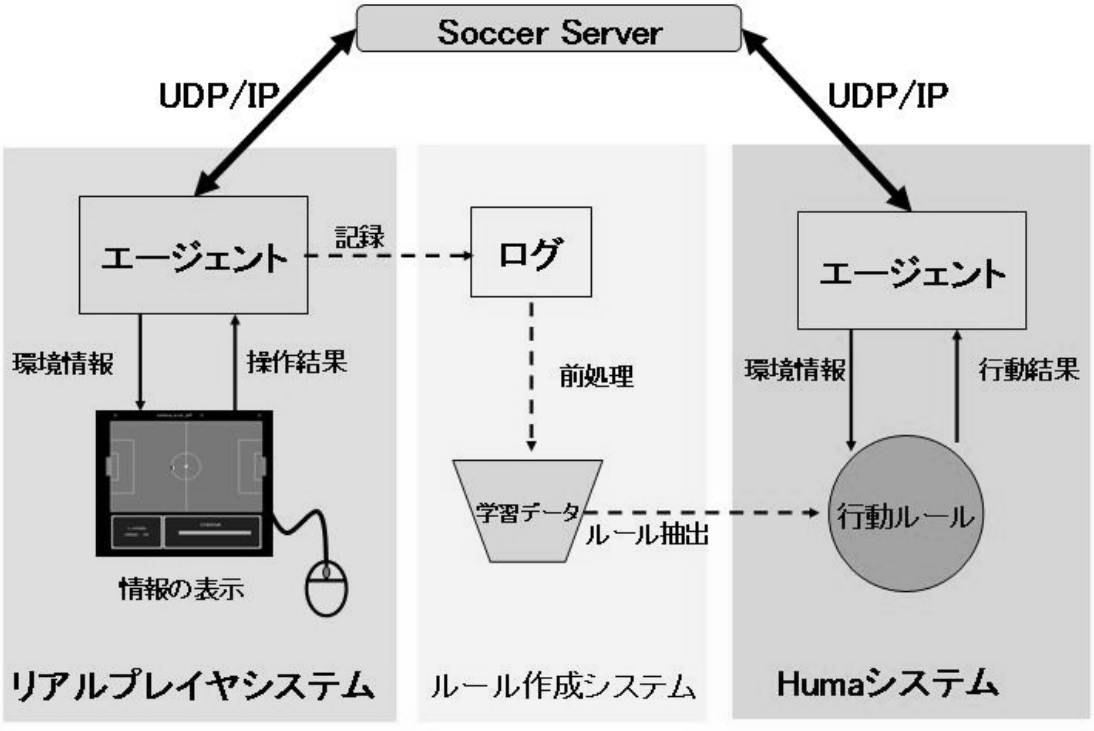

図 3.2　観察による学習システム

3.2.1　リアルプレイヤアーキテクチャ

観察による学習を適用に必要な物として，人間が実際にエージェントと同じ環境で行動を行うためのツールが必要となる．RoboCup シミュレーションでは，人間が意志を決定できるリアルプレイヤシステムが秋田氏らによって提案され，開発が行われている [6]．本稿では，秋田氏らが提案したリアルプレイヤの概念を基に独自に開発したリアルプレイヤシステム [7] を使用した．

リアルプレイヤは，KUT RoboCup Project[8] において開発されたエージェントをベースに開発を行った．KUT RoboCup Project は，高知工科大学情報システム工学科では，「RoboCup シミュレーションを通じてのソフトウェア工学及び人工知能の教育」と題して，RoboCup を使った授業・研究を１９９９年度より行っている．ベースとなったエージェントは，ハンドコーディングにより行動が定義されたエージェントで，2001 年の JAPANOPEN に出場したものを使用した．

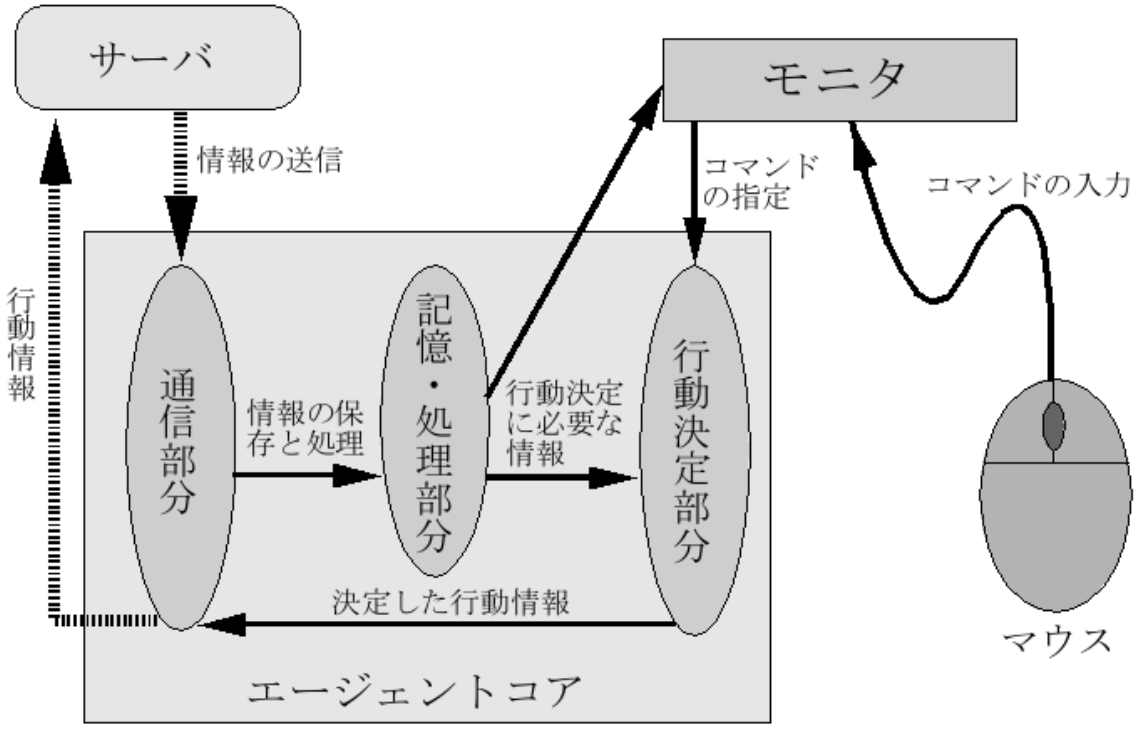

図 3.3　リアルプレイヤ概略図

リアルプレイヤの概略図を図 3.3 に示す.

リアルプレイヤは，大きく分けて，エージェントコア，モニタ，入力ディバイス（マウス，キーボード等）の 3 つに分けられる.

エージェントコアは，

- RoboCup シミュレーションで動かすことのできる最低限の機能
- サーバから得られた情報を加工・格納をする機能
- 複数コマンドによる一連の行動をすることができる機能

を持っている.

RoboCup シミュレーションで動かすことのできる最低限の機能とは，RoboCup シミュレーションで定義されているプロトコルの送受信を行う部分である．通信の確立，コマンドの送信，視覚情報等のセンサ情報の受信，終了処理を行う.

２つ目のサーバから得られた情報を加工・格納をする機能は，サーバから得られた情報を

図 3.4　リアルプレイヤモニタ

基に環境の予測，補間を行う．この機能によって新たに得られる情報は，各物体の絶対座標，絶対方向，見えない物体の予測，物体の将来位置の予測である．

３つ目の基本的な行動とは，人間が１サイクル毎にエージェントへ命令を行うことは不可能である．そこで，複数サイクルに渡る一連の行動を定義しておく．今回のエージェントには，指定した場所へキック，ダッシュによる移動，ドリブル，その場で待機の４つの基本行動を定義した．また，ボールを自動的に追跡するセミオートモードと，完全自動で指示を受け付けない完全オートモード，指示を出さない限り動かない，マニュアルモードが存在する．

モニタは，RoboCup シミュレーションで使われているすべての情報が表示されるモニタとは異なり，エージェントが実際に受け取ったセンサ情報のみを利用して表示される．実際のモニタ画面を図 3.4 に示す．

図 3.4 には，サーバから受け取った情報である自分の位置とボールのおおよその位置，敵や味方などの見えた物のおおよその位置，現在のコマンド，エージェントのスタミナを表示

する．これにより，人間がモニタから実際に得られる情報はエージェントと同じ環境になる．また，モニタは入力ディバイスによる行動を受ける機能を持つ．

入力ディバイスには，3 つボタンマウスとキーボードを使用した．マウスやキーボードによりモニタを通じて各種行動の指定場所を決定し指示を行う．

RoboCup では，3.1 章で説明したように，100msec 毎に 1 つの行動を行うことが出来る．しかし，人間が 100msec 毎に行動を指示することは不可能である．そこで，複合命令を定義した．

- drib 指定場所に，ボールを確保した後ドリブルする．
- dash 指定場所に走って移動する．
- kick ボールを確保した後，指定した場所に向かってボールを蹴る．
- Wait その場で待機する．常にボールの方向を向く．
- Keep ボールを確保し，その場で待機する．
- Auto ボールを確保し，相手ゴールの中心に向かってドリブルする．

これらの複合命令は，それぞれの行動を行うために 1 サイクル毎に，適切な行動を指定する．

複合命令をマウスボタンの右，中，左にそれぞれ，dash，kick，drib, キーボードの “w” を Wait, “q” を Keep, それ以外のアルファベットキーで Auto に割り当てた．

これらの機能に加えて，各サイクル毎のセンサ情報 (自エージェント情報，他エージェント情報，ボール情報)，人間が選択した行動の種類，行動の指定場所のログを採る機能を備えている．ログの例を図 3.5 に示す．

センサ情報に関しては，実際に情報が得られたもののみを記録する．

3.2.2 ルール作成システム

ルール作成システムは，リアルプレイヤで得られた人間の行動のログからルールを生成するシステムである．本システムでは，ルール生成のために C4.5[9] を用いて決定木を作成した．C4.5 は，与えられたデータに対するパターン解析を期待情報量最大化原理に基づいて

3.2　観察による学習システムの構築

図 3.5　ログ例

表 3.1　入力データの属性値名と値

属性名	値
myX	continuous
myY	continuous
myBodyAngle	continuous
ballDirection	continuous
ballDistance	continuous
enemyDirection	continuous
enemyDistance	continuous
friendDirection	continuous
friendDistance	continuous

行なうことにより，帰納的に決定木を作るツールである．

決定木を作成するにあたり，入力するデータと属性を定義した．定義したものを表 3.1 に示す．

属性には上から順番に，エージェントの座標，エージェントの向いている方向，エージェントとボールとの相対角度，ボールまでの距離，一番近い敵エージェントとの距離と相対角度，一番近い味方エージェントとの距離と相対角度を連続値で定義していることを表して

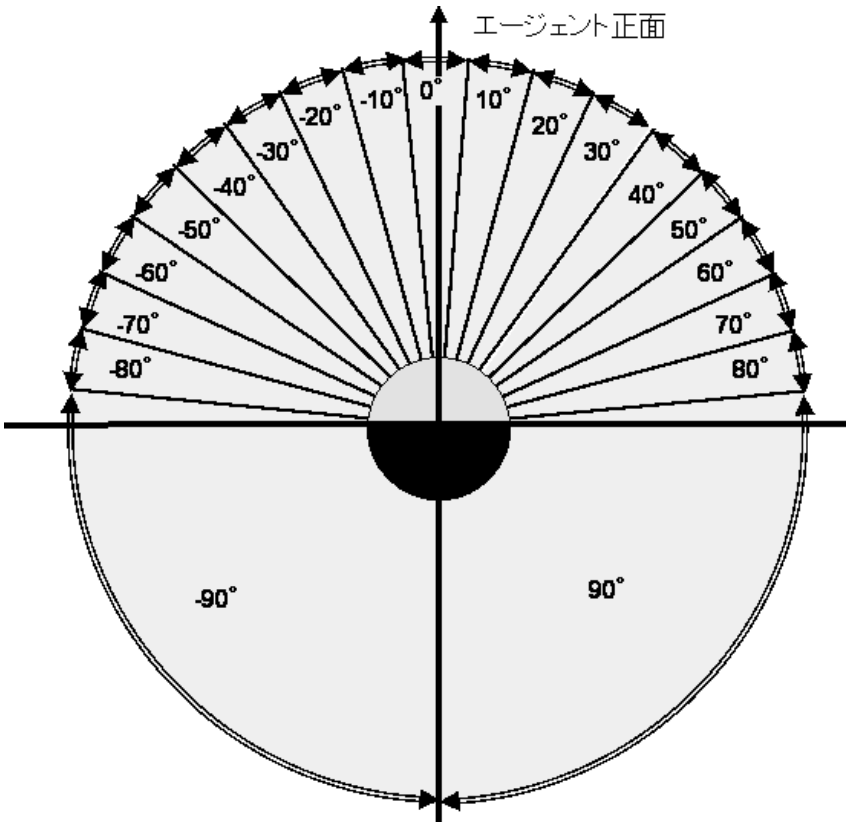

図 3.6　相対角度の定義

いる．

次に，分類のためのクラスの定義である．決定木では，あらかじめクラスを定義しておく必要がある．定義したクラスは，リアルプレイヤで定義した行動をクラスとして使用した．しかし，C4.5 ではクラスは離散的で無ければならないので，行動を離散的に表現することを行なった．

リアルプレイヤの kick, dash, drib の行動のログは，選択した行動と行動を指定した座標が得られる．そのときのエージェントの座標，体の向きと，選択した行動の座標から，エージェントと行動を指定した位置との相対角度を求めた，

求めた相対角度を，離散的にするために１０度刻みで分割した．相対角度の定義を，図 3.6 に示す．

また，kick 行動の指定先が相手ゴールだった場合，指定したゴールの位置によって，

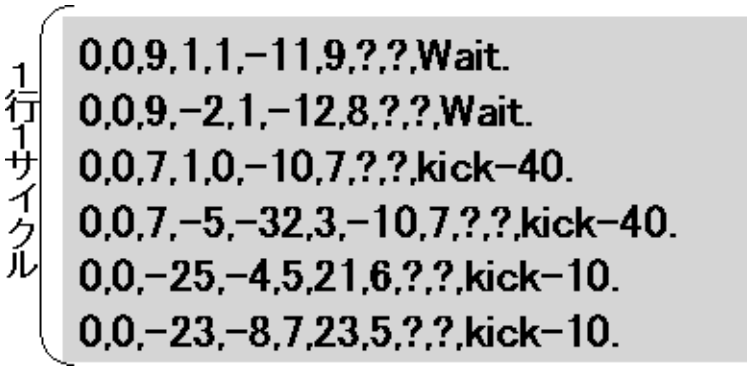

図 3.7　学習データ

ShootTop, ShootMid, ShootBot と定義した．

Wait, Auto, Keep の各クラスは，行動に座標指定が無いため単体で定義した．

定義したクラスを表 3.2 に示す．

定義した，入力値の種類と属性，分類するクラスに基づいた学習データを作成する必要がある．決定木生成の前処理として，リアルプレイヤで得られたログから C4.5 の学習データフォーマットに変換する前処理プログラムを作成した．前処理プログラムから生成される学習データの例を図 3.7 に示す．図 3.7 は，図 3.5 から前処理プログラムによって生成されたものである．

学習データ中の “?” は欠測値を表す．

3.2.3　Huma システム

Huma システムは，ルール作成システムによって生成されたルールを基に行動するエージェントである [10]．Huma システムの概略図を図 3.8 に示す．

エージェントコアの部分は，リアルプレイヤシステムと同じ物を使用している．個となる点は，人間によって決定されていた行動を，生成されたルールによって現在とるべき行動を決定している．

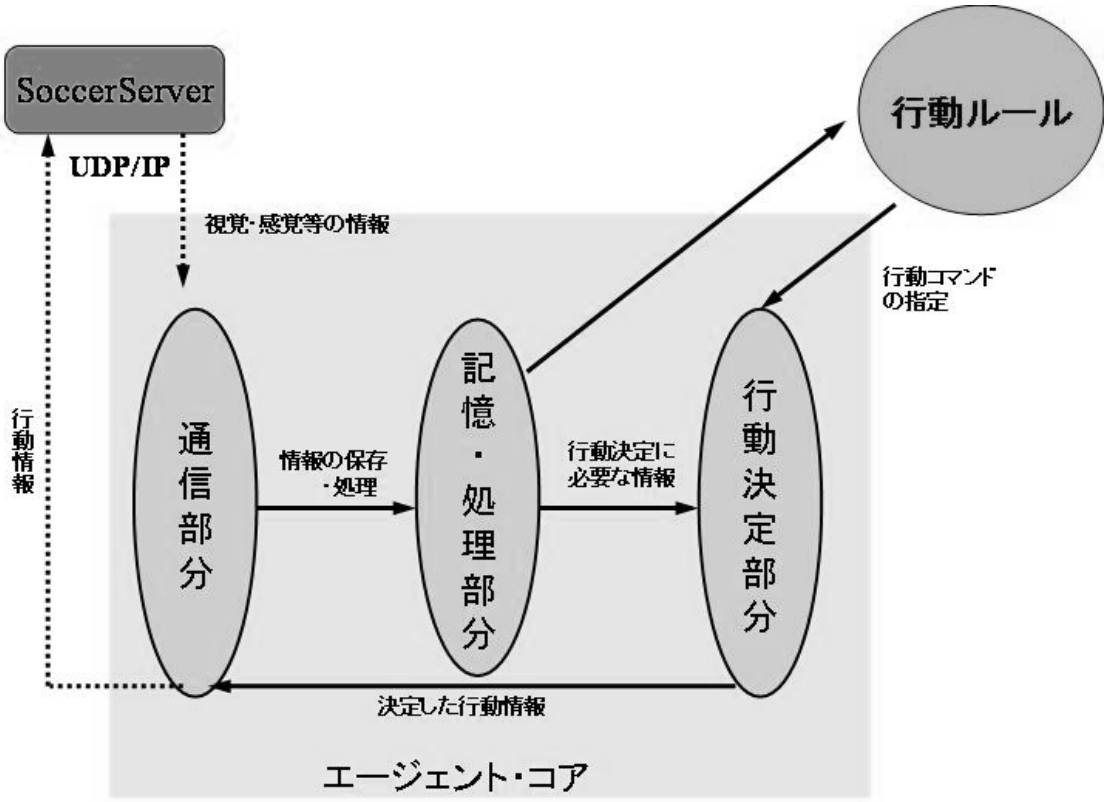

図 3.8　Huma システム

表 3.2　定義したクラスの一覧

クラス名				
kick-90	dash-90	drib-90	ShootTop	Wait
kick-80	dash-80	drib-80	ShootMid	Auto
kick-70	dash-70	drib-70	ShootBot	Keep
kick-60	dash-60	drib-60		
kick-50	dash-50	drib-50		
kick-40	dash-40	drib-40		
kick-30	dash-30	drib-30		
kick-20	dash-20	drib-20		
kick-10	dash-10	drib-10		
kick0	dash0	drib0		
kick10	dash10	drib10		
kick20	dash20	drib20		
kick30	dash30	drib30		
kick40	dash40	drib40		
kick50	dash50	drib50		
kick60	dash60	drib60		
kick70	dash70	drib70		
kick80	dash80	drib80		
kick90	dash90	drib90		

第 4 章

階層化複数決定木の適用

3.2 章で述べたように，本稿では，観察による学習を適用あたり決定木を利用している．しかし今回，分類しなければならないクラスが 63 と多い．決定木では，クラスの数が増えると一般的に認識率が下がる．

そこで，階層化複数決定木を用いる [11][12]．これは，分類するべきクラスを階層的に分割し，分割したクラスを分類する決定木をそれぞれ作成する．これにより，一つの決定木が分類しなければならないクラスの数を減らすことができ，認識率の向上が期待できる．階層化複数決定木の概念図を図 4.1 に示す．

上位クラスの決定木により，次に選ぶべき下位クラスの決定木を決定する．選択された下位クラスで，最終的なクラスを選択する．

階層化複数決定木の観察による学習システムへの適用について述べる．本稿では，分類クラスは表 3.2 で示したように 63 個のクラスがある．これを，階層化を行なうために上位クラスとして，

- kick
- dash
- drib
- Shoot
- Wait
- Auto
- Keep

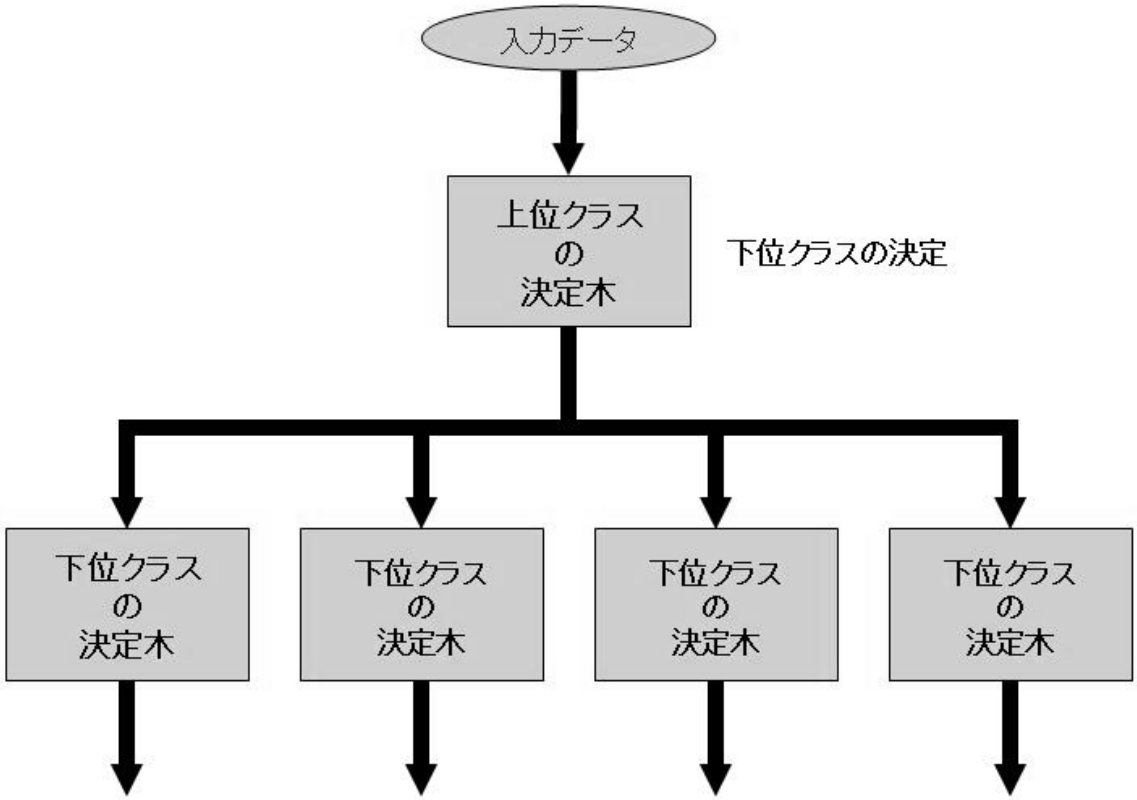

図 4.1　階層化複数決定木概念図

を選択し，これら 7 クラスを分類する決定木を作成する．7 クラスのうち，Wait, Auto, Keep は下位クラスを作る必要がないため，上位の決定木で選択された時点で決定となる．

次に，下位クラスの決定木として

- kick-90 ... kick90
- dash-90 ... dash90
- drib-90 ... drib90
- ShootTop ShootMid ShootBot

のそれぞれ，分類できる決定木を作成する．

これにより，決定木 1 つあたりの分類すべきクラスの数は，上位で 7 クラス，下位で 3〜20 クラスと軽減することが出来る．

また，階層化することで，はじめにおおまかな行動を少ないクラスで分類することから，

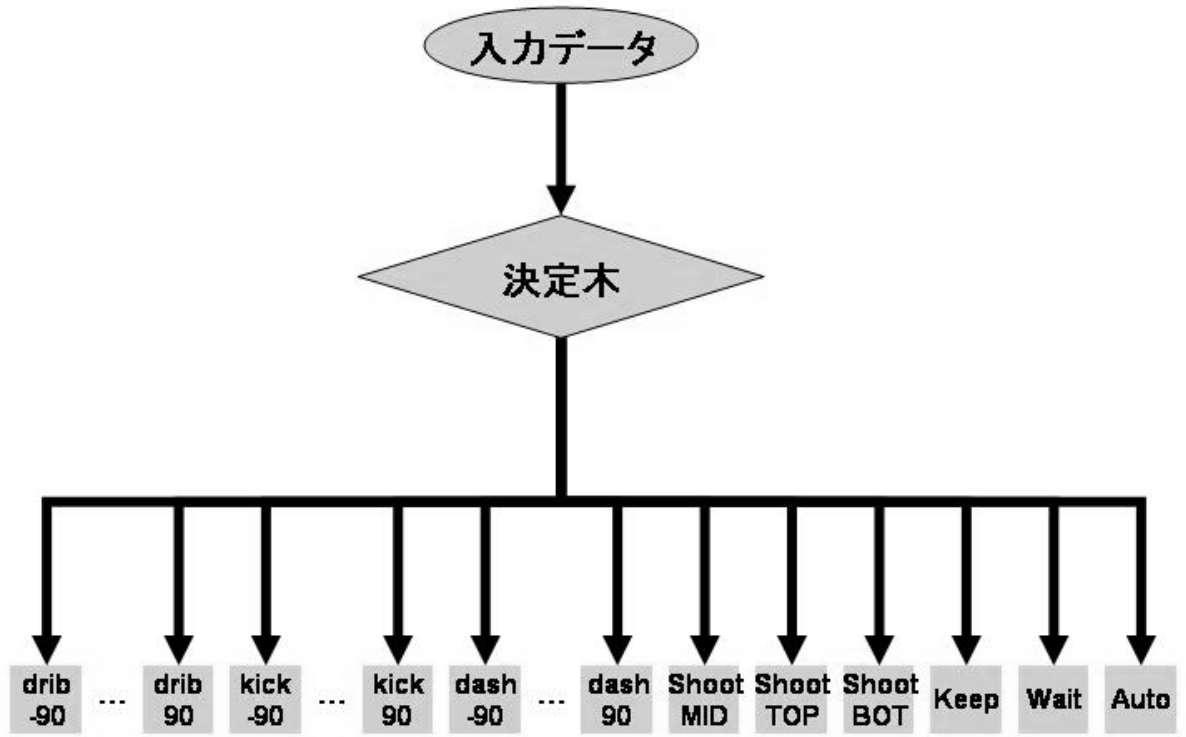

図 4.2　非階層決定木

非階層より全く異なるクラスを選択する可能性が低くなることが期待される.

ここで，本稿における非階層決定木と階層化複数決定木構造を図 4.2 と図 4.3 に示す.

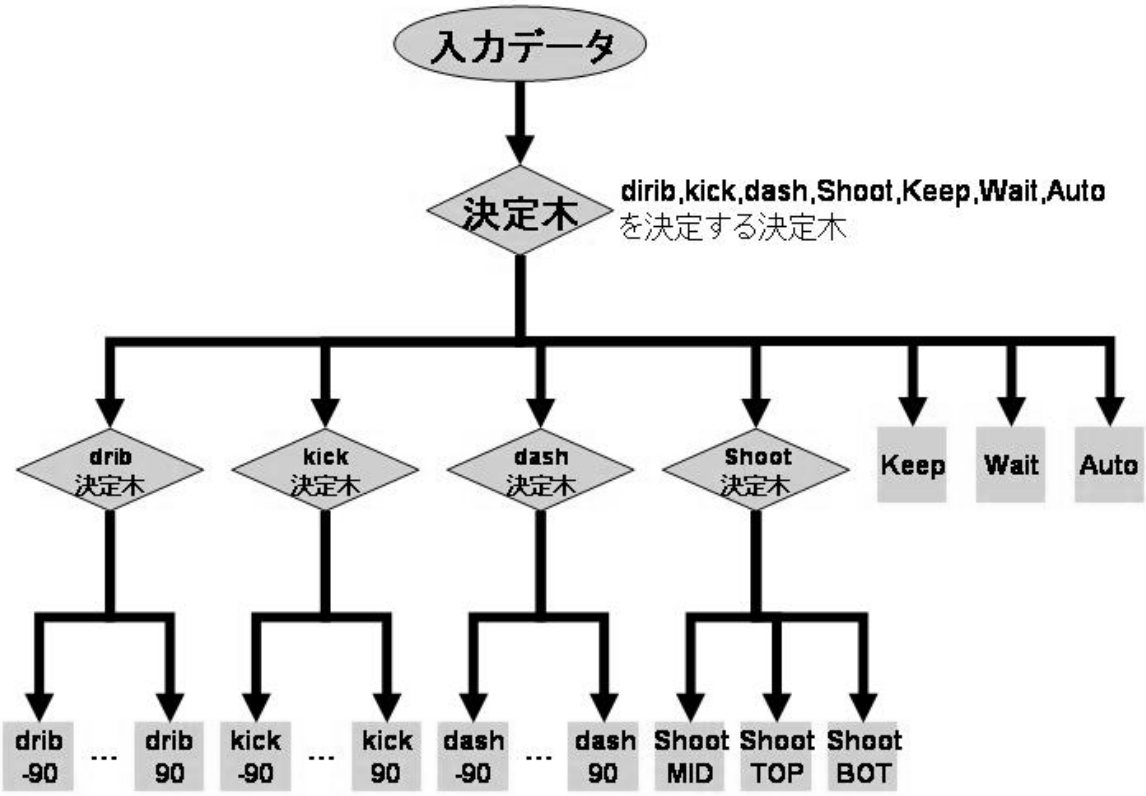

図 4.3　階層化複数決定木

第 5 章

実験

階層化複数決定木を用いた観察による学習の有効性を検証するために以下の実験を行った．

まず，はじめにリアルプレイヤシステムを利用して人間の行動のログを取得する．ログを取得する際に人間が行う行動をあらかじめ定義した．行動は，

1. ボールを持っている MF に DF がボールを取りにくる．図 5.1
2. MF は DF を引きつけたら，ボールを取られないようにパスをだす．同時に FW は，走り出す．この際，オフサイドにならないようにタイミングを図る必要がある．図 5.2
3. 走ってパスを受けた FW は，GK をかわしてシュートする．図 5.3

と定義した．

定義した行動の FW をリアルプレイヤシステムで人間が操作を行う．また，MF, DF, GK エージェントはハンドコーディングによるエージェントを用いる．実際に得点を取ることの出来た事例を 10 事例採取する．

次に，得られたログから非階層決定木と階層化複数決定木を生成する．それぞれの決定木の汎化能力を検証するために，採取した 10 事例のうち 9 事例で決定木を作成し，残り 1 事例をテストデータとする 10 通りの決定木とテストデータを用意する．それぞれの，決定木にテストデータを投入して認識率を評価した．

最後に実際に Huma システムに決定木を適用し，エージェントの行動について評価する．適用する決定木は，採取した 10 事例で階層化負数決定木，非階層決定木を用いる．評価方法は，それぞれの決定木を適用して 20 回行動を行った場合の成功率を検証する．

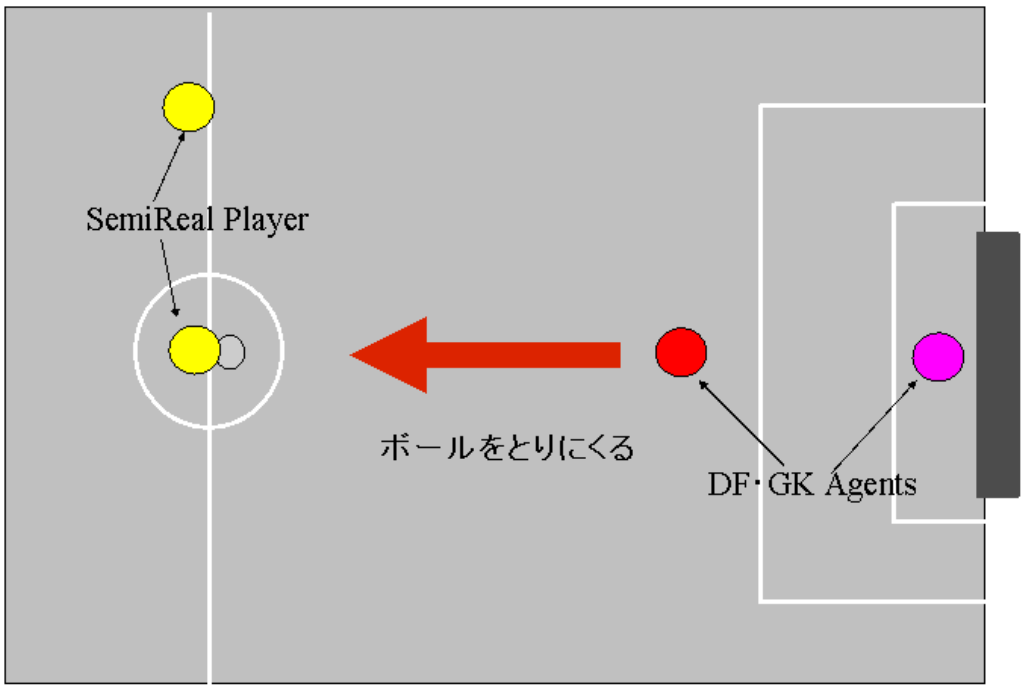

図 5.1　基本戦略 1

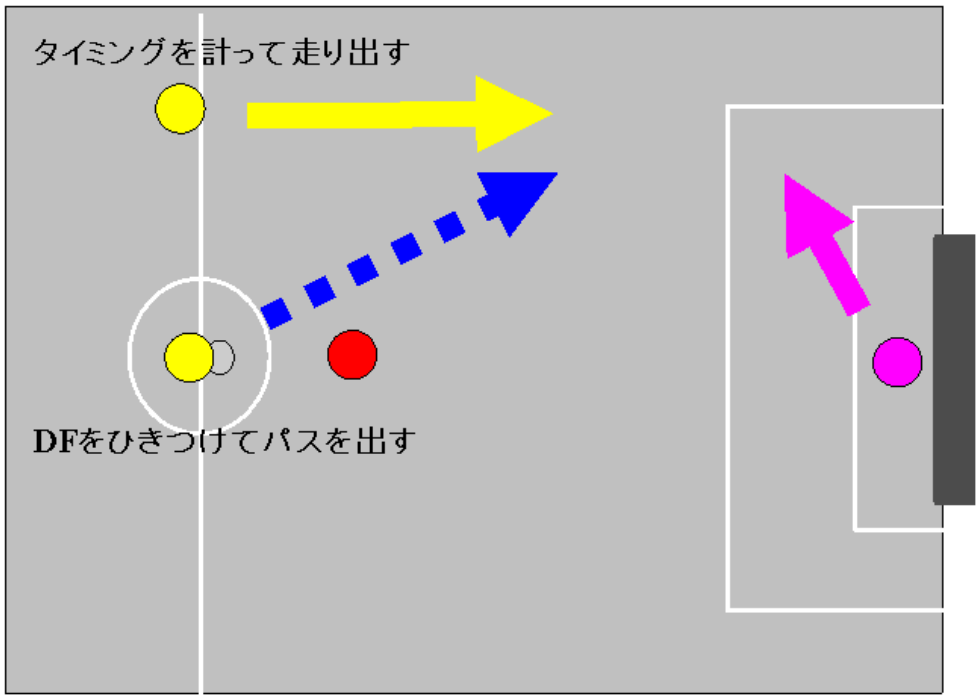

図 5.2　基本戦略 2

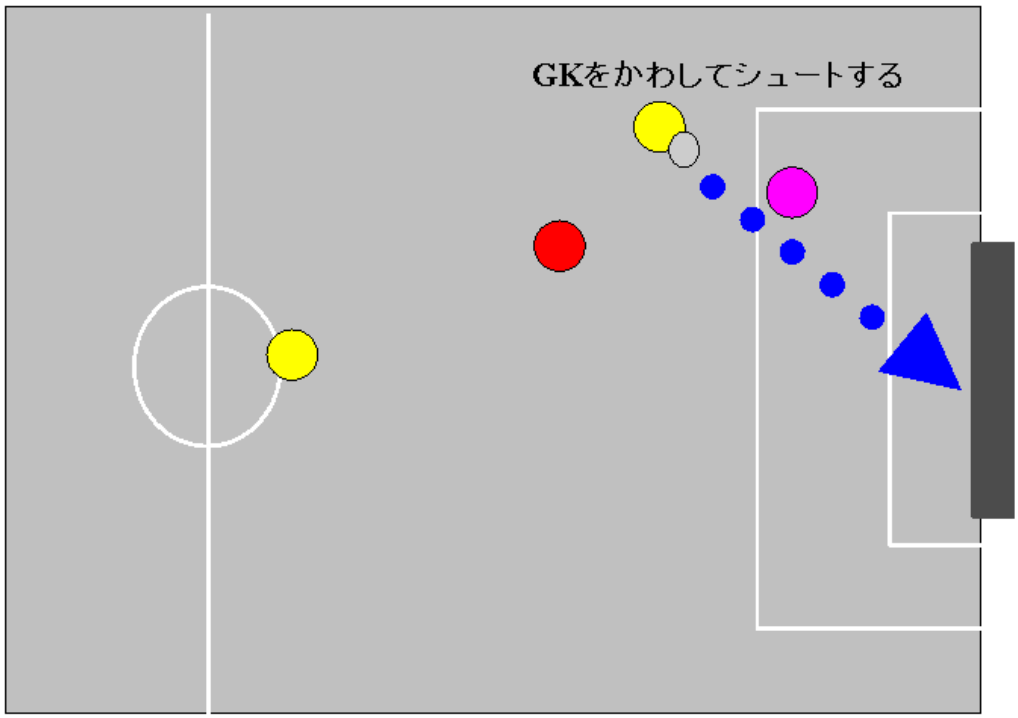

図 5.3　基本戦略 3

第 6 章

実験結果

2 つの実験結果について述べる．

6.1　汎化能力

はじめに，汎化能力についての実験結果を表 6.1 に示す．結果は，10 通りのテストの平均表中の，“認識” はクラスが完全一致．“準認識” は上位クラスまで一致．“誤認識” は不一致を示している．上段は認識率，下段はテストデータの数を示している．全テストデータは，887 個である．認識率は階層化することにより，誤認識率が 24.46%から 21.31%に減少した．

表 6.1　認識率結果

非階層			階層		
認識	準認識	誤認識	認識	準認識	誤認識
57.38%	18.15%	24.46%	62.34%	16.35%	21.31%
509	161	217	553	145	189

このことから，階層化することにより汎化能力が向上することが確認できた．

6.2　エージェントの性能

つぎに，採取した 10 事例で作成した階層化複数決定木と非階層決定木でエージェントを，20 回動かした場合の成功率を表 6.2 に示す．表中では，“成功” は得点した場合，“準成功”

はシュートまで打てた場合, “失敗” はその他の状態を表す.

全体的に，認識率よりも成功率が下がっている．しかし，階層化により行動失敗率が，65%から 50%に低下した．さらに，実際の行動を見ると，階層の方が，非階層よりも人間の行動に近い行動をしていた.

表 6.2　エージェント行動成功率

非階層			階層		
成功	準成功	失敗	成功	準成功	失敗
20.00%	15.00%	65.00%	25.00%	25.00%	50.00%

この結果より，階層化による一定の効果が確認できた.

また，Huma エージェントの行動を向上させるために，決定木作成の前処理として以下の手法を試みた.

- スコープ
  - サッカーフィールドをいくつかのエリア分割して，それぞれのエリアに決定木を作成する.
- クラスの出現頻度の均等化
  - 採取したログから生成される学習データのクラスは，出現頻度にばらつきがあるため，出現頻度を均等化を行う.

しかし，これらの手法を用いても Huma エージェントの行動成功率向上にはつながらなかった.

第 7 章

まとめ

今回の実験によって，汎化能力におうて非階層決定木よりも階層化複数決定木の有効性が確認された．決定木を階層化することにより，誤認識を軽減することが出来る．

また，実際に観察による学習の適用のためにエージェントへ用いた場合でも，非階層決定木使用した場合よりも階層化複数決定木の方が成功率は向上した．

しかし，エージェントでの成功率は決定木そのものの認識率より低い結果だった．原因として，階層化，非階層ともに生成されたルールが影響していると考えられる．

生成されたルールのすべてにおいて，初めに自エージェントの座標を調べる．また約半数のルールが，ボールや他のエージェントの情報を調べることなく，自エージェントの情報のみでクラスを決定している．自エージェント以外の情報を見るルールと見ないルールの数を表 7.1 に示す．階層の方は，上位クラスの決定木のルールのみを示す．これは，自エージェントの情報は確実に毎サイクル得られ，ボールや他のエージェントの情報は欠測となることが多いためと考えられる．

表 7.1　生成されたルールにおける自エージェント以外の情報の使用有無

非階層			階層		
他を見ない	他を見る	ルール合計	他を見ない	他を見る	ルール合計
26	18	44	12	8	20

この結果，エージェント自身の場所と状態により行動が決められてしまい，敵の位置を見てタイミング良く走り出す等の行動がうまく行なえていないと考えられる．

この問題を解決するために，学習データの入力属性に重みをつける等をおこない，周りの環境から結果を導き出す手法を検討する必要がある．

また他の原因として，リアルプレイヤで成功事例を採取したときの，

- 同じ状況でも時でも違う行動を選択してしまう人間の曖昧さ
- 人間の判断の遅れ
- エージェントが得られていない情報を人間が予測し補完して行動するため，予測の部分が決定木に反映されない

影響などが考えられる．

1 つ目の人間の曖昧さに対応するために，出現頻度の高い方で対応できると思われる．

2 つ目の人間の判断の遅れに対応する方法として，RoboCup のシミュレーション時間を遅くすることにより，人間が余裕をもって判断出来るようにする方法が考えられる．

3 つ目の予測の部分が決定木に反映されない問題は，エージェントコアの機能の一つである予測機能向上，あるいは，人間が予測した物を記録するツールを作成し反映させることが考えられる．

今後の課題として，人間の判断を的確にエージェント伝えることの出来るようにリアルプレイヤの改良が挙げられる．また，エージェントの行動成功率向上，より複雑な行動に対応するために，階層化複数決定木を作成するログの前処理方法の検討が必要である．

謝辞

本研究を進めるにあたり，大変丁寧な御指導をしてくださった，立命館大学理工学部情報学科の Ruck THAWONMAS 助教授に感謝致します．また，突然の配属にも快く受け入れて頂いき，さらに手厚い指導をしてくださった，本学情報システム工学科の竹田史章教授にお礼を申し上げます.

実験, プログラムの実装に関して協力していただいた，竹田研究室の日野 慎一君に感謝する．また，日頃の学校生活を有意義かつ，楽しいものにしてくれた，旧ラック研究室のメンバー，竹田研究室のメンバー，友人に感謝します.

参考文献

[1] 長尾　確，エージェントテクノロジー最前線，共立出版 (2000)

[2] http://www.robocup.or.jp/

[3] Richard S. Sutton, Andrew G.Barto, “強化学習，” 強化学習，三上 貞芳（共訳），皆川 雅章（共訳），pp.3–5，森北出版株式会社，東京，2000.

[4] Y. Kuniyoshi, M. Inaba, and Inoue, “Learning by watching:Extracting reusable task knowledge from visual observation of human performance,” In IEEE Transactions on Robotics and Automation, pp.799–822, 1994.

[5] 平山 純一郎, Ruck Thawonmas,“RoboCup ソフトウェアエージェントへの人間の意思決定挙動の適用”，電子情報通信学会「人工知能と知識処理」11 月研究会, 信学技報 AI2001-54, pp. 57-61, November. 2001.

[6] 秋田純一， 西野順二， 久保長徳，下羅弘樹， 藤墳到，“RoboCup シミュレーションリーグ人間参戦システム OZ-RP の提案”, 人工知能学会第１２回 Sig-Challenge 研究会資料，pp.23–28, April. 2001.

[7] 平山 純一郎，田村 和也，日野 慎一，川口 宏，Ruck THAWONMAS，竹田 史章，“人間挙動観察によるエージェント学習システム”, 情報処理学会四国支部シンポジウム，pp.11–14, March. 2002.

[8] Ruck Thawonmas，大森 洋一，平山 純一郎, “RoboCup ソフトウェアロボットづくりによる問題設計解決型学習”, 第９回情報教育方法研究発表予稿集，pp.50–51, July. 2001.

[9] J.R. Quinlan，AI によるデータ解析，古川 康一（監訳）,（株）トッパン，東京，1995.

[10] R. Thawonmas, J. Hirayama, and F. Takeda, “Learning from Human Decision-Making Behaviors - An Application to RoboCup Software Agents, ” IEA/AIE 2002, Cairns, Australia, June 2002.

[11] 平山 純一郎, Ruck THAWONMAS, 竹田 史章,“ 階層化した複数決定木を利用した観察による学習とその RoboCup への適用 ”情報処理学会四国支部シンポジウム, pp.101–106, March. 2002.

[12] Ruck Thawonmas, Junichiro Hirayama, and Fumiaki Takeda, “RoboCup Agent Learning from Observations with Hierarchical Multiple Decision Trees, ” PRIMA2002, pp.7–15, August. 2002.