卒 業 論 文

実世界のエージェント構築のための
シミュレーション環境の実現

奈良女子大学理学部情報科学科 4 回生 新出研究室
学籍番号　11251841

柚木静香

概要

実世界において自律的に振る舞うロボットの実現に向けては様々な実験が必要となるが、それらをいきなり現実世界で実験するのは困難であるので、仮想世界でのシミュレーションが有用である。そこで、より実世界に近い環境を構築するためのシミュレーションの作成を目指した。

亀村・林による研究 [3, 1] では、例題としてカヌーレーシングを用いており、エージェントが学習を行って獲得した行為を用いて自律的に行動することが可能になったものの、プログラム全体がカヌーシミュレーションの例に依存したものであった。従って本研究では、一般的なシミュレーションプログラムとカヌーシミュレーション部分とを分離し、多様な環境および問題に対応可能なシミュレータの実現を目指した。本論文では、クラス設計と実装について述べる。

1 はじめに

BDI モデルとは、信念（B)、願望（D)、意図（I) と呼ばれる３つの心的状態を用いて意思決定を行うモデルである。我々は、自律型エージェントの実装を目指すために、BDI モデルを用いた合理的かつ自律的なエージェントを用いたシミュレータの研究を行ってきた。

実世界で自律的に振る舞うロボットの実現においては、実世界で生じる様々な問題に対する解決策を導き出す必要があるが、いきなり現実世界で実験することは困難である。様々な実験をより短期間で行うには仮想世界におけるエージェントのシミュレーション環境が有用であると考え、我々はより実世界に近い環境下で動く、柔軟性の高いシミュレーションの作成を目指した。

先行研究 [3, 1] では、Jason とよばれる言語処理系と Java を用いて、カヌーレーシングをテストベッドとしたシミュレーション環境の作成を行った。また、エージェントに自律的に自己の行為を選択させるため、エージェントの知覚とその知覚を用いて行動を獲得するための学習機構の追加に成功した。しかし、従来のカヌーレーシングのシミュレーションは、川の形状や流れに関しては予め定められたものを使用しており、一定の環境状態での実験しか行うことができない状態であった。そのため、カヌーレーシングのシミュレーションにおいて川環境による難易度の設定ができなかった。また、プログラム全体が、例として選んだカヌーレーシングのシミュレーションに依存したものであり、柔軟性に欠けていた。更にその他の問題点として、強化学習の学習方式が固定であった。

そこで我々は、従来のプログラムが一般的なシミュレーションプログラムとしてより様々な実験を行うことを可能にするために、クラスの設計の変更によりプログラムの柔軟性を上げ、汎用化を図った。また、強化学習においては複数の学習方法の実装を行い、カヌーレーシングでの最適な学習方法の決定を目指した。

本研究は奈良女子大学理学部情報科学科４回生宮田との共同研究である。本論文ではクラス設計と実装について述べる。強化学習の実験と結果については [4] で述べられている。

2　プログラムの構造

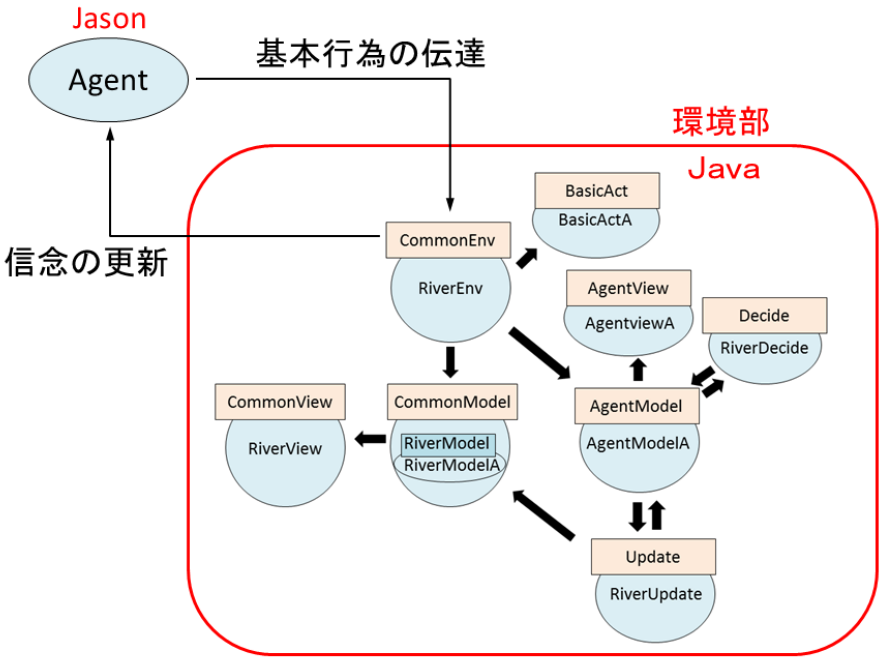

図 1　プログラムの構造

プログラム構造は図 1 のようになっている。プログラムの作成において、エージェントの記述には BDI モデルを用いたエージェントを構築するためのプラットホームである Jason を、環境の設定には Java を用いた。図 1 の赤線枠内は環境部であり、本論文ではこの部分の設計の変更点について述べる。

3　カヌーシミュレーション実装方針

3.1　実装方針 1

カヌーレーシングのシミュレーションプログラムにおいて、従来の研究では川の形状に関しては予め定められた一定の環境状態のみで実験が行われていたので、川の形状などの一部の環境設定を変更することによってシミュレーションの難易度を変化させることに対応していなかった。そこで本研究は、川の形状（形、幅、長さ）の変更を容易にすることを目指した。

問題解決に向けて実際に行ったクラス設計の変更点は以下のとおりである。

1. 川環境に関する設定を行うクラスの統合
2. エージェントと川環境の描画クラスの分離
3. 川の設定を行うクラスと基本行為の処理を行うクラスの分離
4. 川の環境を設定するクラスである RiverModel の子クラスの作成

2 および 3 の分離によりエージェントと川環境のクラスが分離されることになり、それぞれのクラスが独立に交換可能になった。1 および 4 のクラス設計の変更により、元々「川の流速の処理」と「川の形状の設定」を独立した別々のクラスが行っていたのを、RiverModel が「川の流速の処理」を、RiverModel の子クラスとして新たに作成した RiverModelA(あるいは RiverModelB) が「川の形状の設定」を行うように変更した。このように親子関係のクラスである RiverModel とその子クラスで川環境の設定を行うように変更することで、具体的な設定値 (川の幅、長さ、形状) は子クラス単位で変更可能になり、各クラスのインスタンス化を行うクラスである RiverEnv のインスタンス化する RiverModel の子クラスを変更するだけで川の形状を容易に変化させることができるようになった。なお各クラスの概要については第 4 章で詳しく述べる。

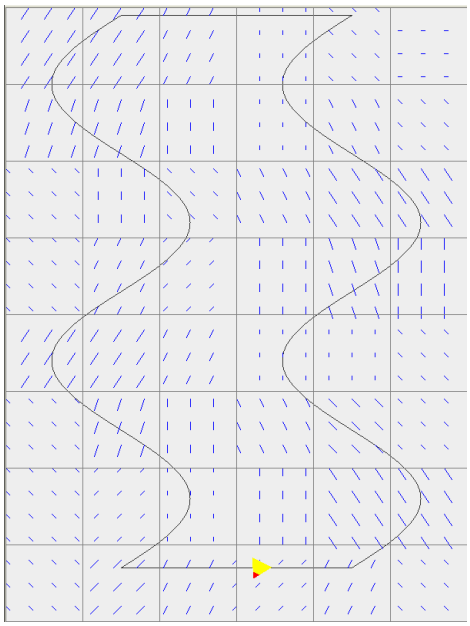

図 2　RiverModelA の描画

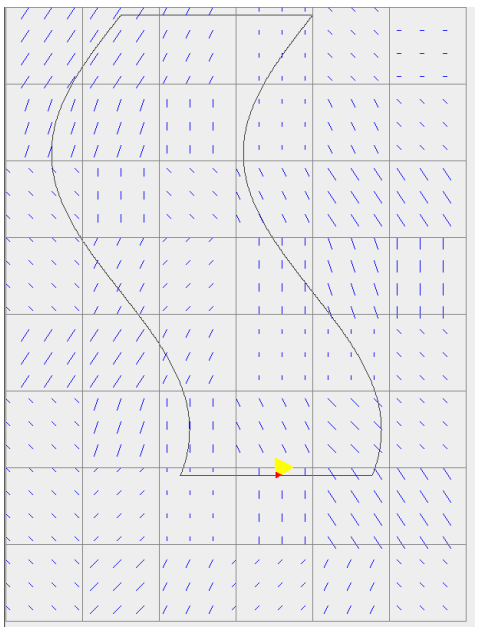

図 3　RiverModelB の描画

3.2　実装方針 2

本研究のシミュレーションプログラムでは、将来的にカヌーレーシングだけでなく、様々な題材を使用したシミュレーション環境の実現を目的としているが、[1, 2] ではプログラム全体が例題として選んだカヌーレーシングのシミュレーションに依存しており、一般的なシミュレーションプログラムとしては柔軟性に欠けていた。そこで、本研究ではプログラム全体のクラス設計を見直し、カヌーレーシングに依存している部分と一般的なシミュレーションプログラム部分に分離した。その結果、クラス設計は以下の図 4 のようになった。黒い矢印は作業依頼、情報伝達の関係を示している。

なおクラス設計の際にデザインパターン [2] を導入し、プログラムの変更容易性の向上を図った。図 4 の緑の枠はデザインパターンの一つである Template Method パターンを使用したことを示しており、親クラスには抽象メソッドを用意し、具体的な実装は子クラスで行うというようにすることで処理の枠組みと実装を分離した。

従来取り扱っていた例題としてのカヌーレーシングシミュレーションに特有な部分を新たに追加した一般的なシミュレーションプログラムのクラスの子クラスとして実装した。従って、カヌーレーシングのプログラム以外のシミュレーションを実験する場合は子クラスのみを取り替えれば良い。

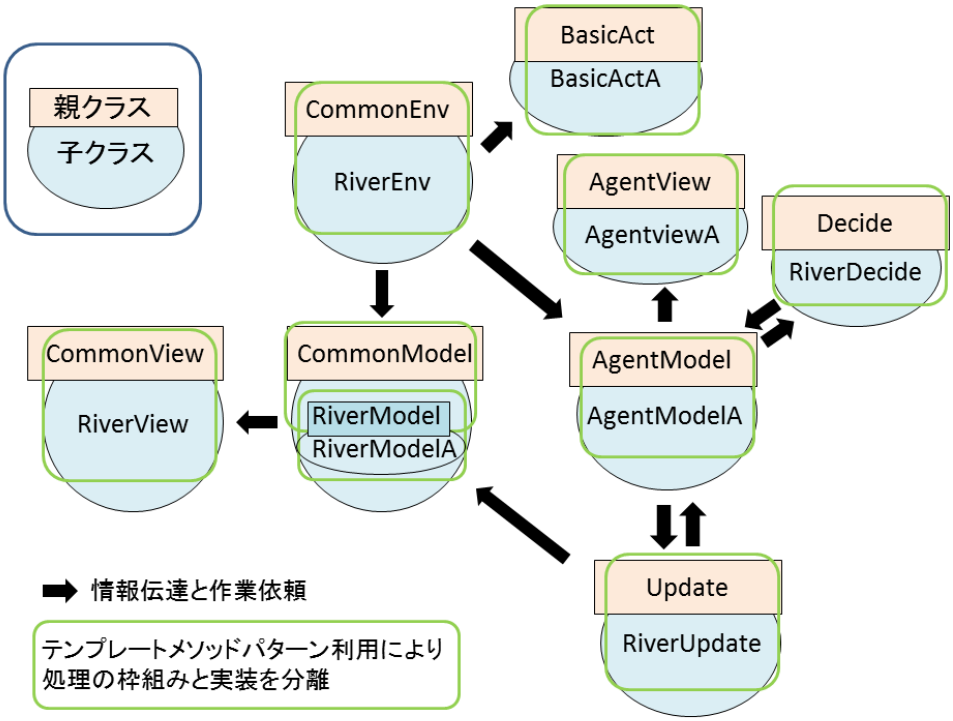

図 4　クラス設計

3.3 実装結果

実装方針 1 および実装方針 2 において、親クラスと子クラスの役割分担により各クラスの独立性が高まり、部品としての交換可能性が高まった。以下に、実際に変更が容易になった点と変更を行う際の留意点についてまとめる。

- RiverModel の子クラスの交換による川の形状に関する変更
  川の形状の変更容易性については第 3.1 項の実装方針 1 で述べたので省略する。

- CommonModel の子クラス (RiverModel) の交換による環境の設定の変更
  CommonModel クラスでは環境の具体的な内容を設定している。本研究で CommonModel の子クラスとして実装した RiverModel は、川の流速の向きおよび速さを設定している。従って川の流速の設定値を変えるには RiverModel クラスを変更する必要がある。このように具体的な環境の設定を CommonModel の子クラスが行うことで、結果として子クラスのみの変更でシミュレーション環境の変更が可能になった。

- BasicAct の子クラス (BasicActA) の交換による基本行為の変更
  BasicAct は基本行為の抽象クラスなので、具体的な行為の内容は BasicAct の子クラスで設定する。よって BasicAct の子クラスを交換することで基本行為の内容を変更することが可能である。
  本研究で BasicAct の子クラスとして実装した BasicActA は、川の形状の変更に影響されることなく使用可能である。カヌーレーシングのシミュレーションにおいて、川の具体的な形状の変更と基本行為の変更は独立に可能であり、それぞれの具体的な内容は子クラスで実装すれば良い。なお、シミュレーションする問題を変更する場合は当然基本行為も入れ替える必要がある。また、基本行為はメソッド毎に一つの行為が実装されているため、基本行為の種類を増やす際には親クラスの BasicAct にも抽象メソッドを追加する必要がある。基本行為をメソッドではなく配列を使用して用意することが出来れば、基本行為の数を変更する際に親クラスを変更する必要がなくなるので、これは今後の課題点である。

- Update の子クラス (RiverUpdate) の交換による強化学習の変更
  Update クラスでは、学習の抽象メソッドおよび強化学習において必要となる評価値計算のメソッドが用意されているため、子クラスでは具体的な学習方式や強化学習の具体的な流れを設定し、親クラスの評価値計算メソッドを使って学習の状態行動価値の推定値を計算する。そのため、学習方式を変更するには子クラスのみを変更すれば良い。
  本研究で Update の子クラスとして実装した RiverUpdate を交換することで変更が可能な点は、強化学習の学習方式や行動選択法である。学習方式の例として Q 学習や Sarsa が挙げられる。また、行動選択法としては、ϵ -greedy 行動選択やソフトマックス行動選択が挙げられる。その他、強化学習の一つのエピソード内の具体的な流れや報酬値の大きさも変更が可能である。しかし、学習機構を持つクラスの中で状態行動価値の推定値を計算する際に、状態が xy 座標による表現に固定されているので、今後の研究で強化学習の精度を高めるにあ

たって改善していく必要がある。

4 クラス概要

4.1 クラスの変更点

従来のカヌーシミュレーションプログラムのクラスである RiverEnv、RiverModel、RiverModel、RiverView、RiverUpdate、RiverDecide は一部を改良・追加した。また RiverAgent は River 依存の部分を切り離し AgentModel の子クラスの一つとしたため AgentModelA と改名した。川の流速の処理と川の環境の設定は従来別々のクラスが行っていたが、川をモデル化する新たなクラス RiverModel に統合した。BasicActA、AgentViewA、RiverModelA、RiverModelB は新しく作成したクラスである。その他、抽象クラスとして CommonEnv、CommonModel、CommonView、AgentView、AgentModel、Update、Decide、BasicAct を追加し、それらの子クラスとなる具象クラスでカヌーレーシングシミュレーションを実装した。

また、従来の研究ではメソッドやフィールドを参照する際に各クラスで必要なクラスのインスタンス化が別々のタイミングで個別に行われていたため、必要なオブジェクトが既にインスタンス化されているかどうかをプログラム中で把握するのは困難であった。本研究では、RiverEnv での初期化処理で必要なオブジェクトをインスタンス化するように変更し、その問題を解決した。

4.2 各クラスの役割

各クラスの概要は以下の通りである。

- CommonEnv
  Jason が提供する Environment クラスの子クラスである。Jason とやりとりをするため、エージェントの基本行為の伝達に関する抽象メソッドを持つ。基本行為を Jason から受け取る。

- CommonModel
  環境の設定を行うクラスである。環境の抽象メソッド持つ。

- CommonView
  描画を担当する抽象クラスである。ウィンドウと格子の描画の設定はここで行う。また、エージェントと環境の描画メソッドを呼び出す。

- AgentView
  エージェントの設定、描画を担当するクラスである。またエージェントの座標の更新と表示を行う。

- AgentModel
  エージェントの位置情報、学習結果を持つクラスである。Update による強化学習の結果と Decide による最適な行為を取得する抽象メソッドを持つ。これらのメソッドは RiverEnv によって呼び出される。AgentModel は Update と Decide を仲介する役割を担っている。

- Update
  学習機構を担当するクラスである。強化学習の学習方式の設定に関する抽象メソッドを持つ。また学習でする評価値計算のメソッドを持つ。学習結果の更新・表示を行う。

- Decide
  AgentModel からエージェントの位置と学習の結果を受け取って、最適な行為を選択するクラスである。

- BasicAct
  基本行為の抽象クラスである。具体的な処理内容は子クラスで実装する。

- RiverEnv
  CommonEnv の子クラスである。CommonEnv が Jason から受け取った基本行為を、BasicAct に伝達して処理する。エージェントが信念として自身の位置情報を持つため、信念を更新するメソッドを持つ。またエージェントが川の中にいるかを判定し、知覚の更新を行う。Java のプログラムの Main 文の役割を果たしている。

- RiverModel
  CommonModel の子クラスで、川の環境の設定を行うクラスである。また川の流速、各地点の位置情報を持つ。川の形状を返すメソッド、与えられた座標値点の川の流速を返すメソッド、エージェントの位置が川の中であるかどうかを判定するメソッドを持つ。

- RiverModelA
  RiverModel の子クラスで、川の形状、幅、長さの設定を行うクラスである。

- RiverView
  CommonView の子クラスで、川の描画を担当するクラスである。

- AgentViewA
  AgentView のサブクラスで、AgentView の River 依存部分の処理を扱うクラスである。

- AgentModelA
  AgentModel の子クラスで、エージェントの位置の初期値を設定する。Update に任意の回数の学習を依頼するメソッドを持つ。また、RiverUpdate による強化学習の結果を持ち、RiverDecide によって導き出された行動推定価値が最も高い行動を取得するメソッドを持つ。

- BasicActA
  BasicAct の子クラスで、基本行為の具体的な処理の設定を行うクラスである。基本行為を受け取ると、AgentModel からエージェントの位置を取得し、その位置 の川の流速からエージェントの速度と流速の合力から次の地点を計算する。そしてエージェントの新しい位置を AgentModel に渡し、最後に RiverEnv に基本行為の実行結果を返す。

- RiverUpdate
  Update の子クラスで状態、行為及び知覚を受け取って漕ぎ方の学習を行うクラスである。学習中の基本行為は BasicAct に依頼し、RiverModel からエージェントの位置情報を取得しゴールかどうかの判定等を行う。学習結果は AgentModel に渡す。Update で実装されなかった具体的な学習方式の設定や強化学習の流れについての設定を行う。

- RiverDecide
  Decide の子クラスで Decide の River 依存部分の処理を行うクラスである。

4.3 クラス図

クラス設計は図 5 のようになっている。この図は一般的な UML 図の書式に従っており、おのおののクラスは長方形で表現し、長方形の中は水平線で分割して、

- クラスの名前
- フィールドの名前
- メソッドの名前

が順番に書かれている。

白抜きの三角がついた実践の矢印はクラスの階層関係を表す。矢印は子クラスから親クラスへ向かっている。

白抜きのひし形がついた線は集約の関係を表す。ひし形と接しているクラスが「持っている」側、線の終点と接しているクラスが「持たれている」側である。

abstract クラス・abstract メソッドの名前は斜字体で書かれている。

static フィールド・static メソッドの名前には下線をつけている。

5 強化学習

従来の研究では強化学習に Q 学習方式のみが用いられていた。本研究では、Q 学習と Sarsa の２つの方式を用いながら、学習によるデータ分析を行った。強化学習の概要や実験結果については [4] で述べる。

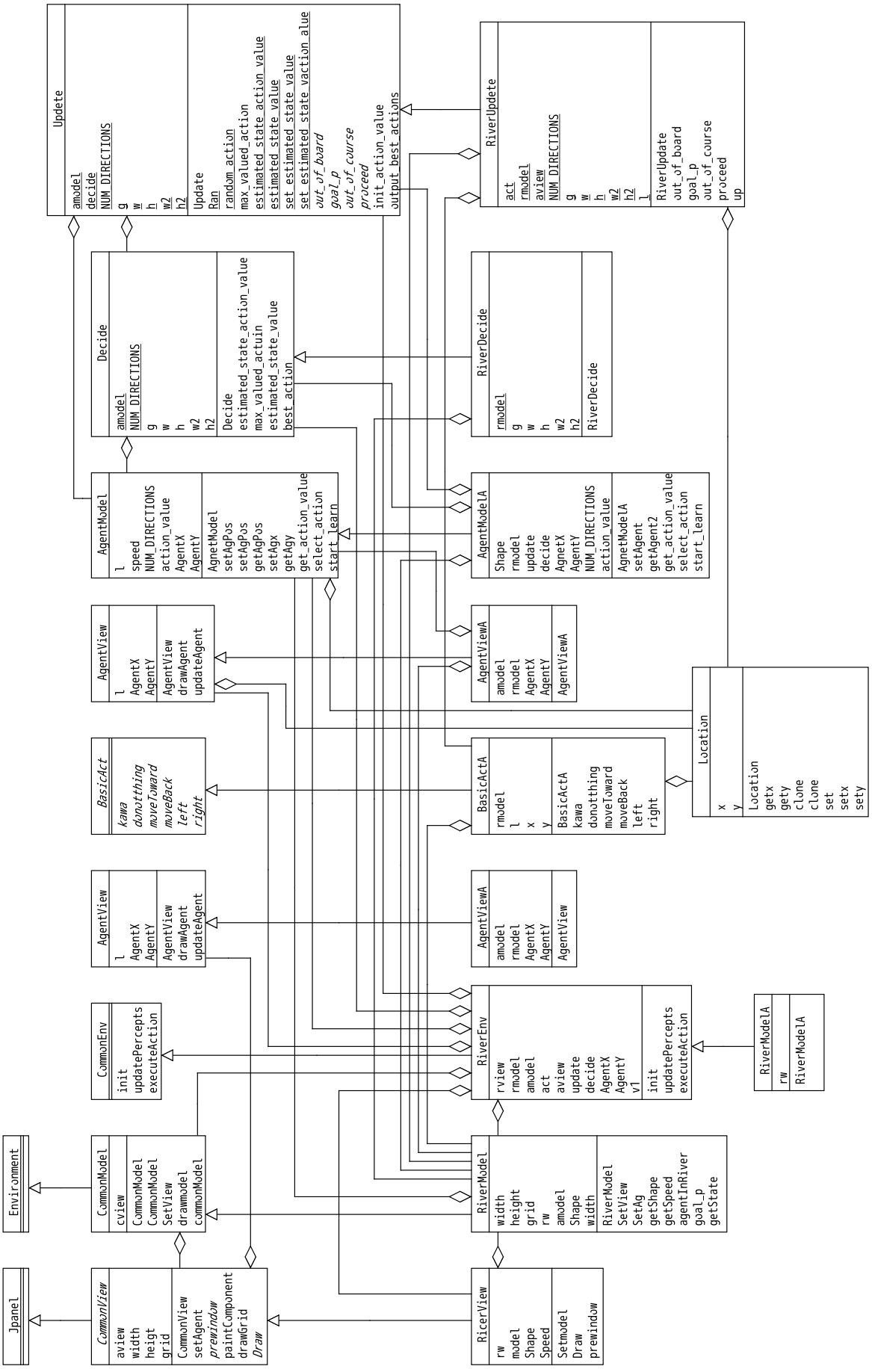
Updete
amodel
decide
NUM_DIRECTIONS
g
w
h
w2
h2
Update
Ran
random_action
max_valued_action
estimated_state_action_value
estimated_state_value
set_estimated_state_value
set_estimated_state_vaction_alue
out_of_board
goal_p
out_of_course
proceed
init_action_value
output_best_actions
RiverUpdete
act
rmodel
aview
NUM_DIRECTIONS
g
w
h
w2
h2
l
RiverUpdate
out_of_board
goal_p
out_of_course
proceed
up
Decide
amodel
NUM_DIRECTIONS
g
w
h
w2
h2
Decide
estimated_state_action_value
max_valued_actuin
estimated_state_value
best_action
RiverDecide
rmodel
g
w
h
w2
h2
RiverDecide
AgentModel
l
speed
NUM_DIRECTIONS
action_value
AgentX
AgentY
AgnetModel
setAgPos
setAgPos
getAgPos
setAgx
getAgy
get_action_value
select_action
start_learn
AgentModelA
Shape
rmodel
update
decide
AgnetX
AgentY
NUM_DIRECTIONS
action_value
AgnetModelA
setAgent
getAgent2
get_action_value
select_action
start_learn
AgentView
l
AgentX
AgentY
AgentView
drawAgent
updateAgent
AgentViewA
amodel
rmodel
AgentX
AgentY
AgentViewA
BasicAct
kawa
donotthing
moveToward
moveBack
left
right
BasicActA
rmodel
l
x
y
BasicActA
kawa
donotthing
moveToward
moveBack
left
right
Location
x
y
Location
getx
gety
clone
clone
set
setx
sety
AgentView
l
AgentX
AgentY
AgentView
drawAgent
updateAgent
AgentViewA
amodel
rmodel
AgentX
AgentY
AgentView
CommonEnv
init
updatePercepts
executeAction
RiverEnv
rview
rmodel
amodel
act
aview
update
decide
AgentX
AgentY
v1
init
updatePercepts
executeAction
RiverModelA
rw
RiverModelA
Environment
CommonModel
cview
CommonModel
CommonModel
SetView
drawmodel
commonModel
RiverModel
width
height
grid
rw
amodel
Shape
width
RiverModel
SetView
SetAg
getShape
getSpeed
agentInRiver
goal_p
getState
Jpanel
CommonView
aview
width
heigt
grid
CommonView
setAgent
prewindow
paintComponent
drawGrid
Draw
RicerView
rw
model
Shape
Speed
Setmodel
Draw
prewindow

図 5　クラス図

6　まとめ

本研究により、一般的なシミュレーション部分をプログラムの枠組みとすることで、例としてのカヌーレーシングのシミュレーションに依存しない、より柔軟性の高いプログラムになった。これにより、カヌーレーシング以外のシミュレーションとしても動かすことが可能になった。また、各クラスの独立性が高まり、プログラムとしての変更容易性、保守性、再利用性が向上し、連続世界を想定した環境下でより一般的なシミュレーションが可能となった。更に、カヌーレーシングのシミュレーションとしては環境や設定値の変更が容易になり、様々な実験が行えるようになった。

今後は、以下の問題について取り組むことが課題である。

- カヌーレーシングのシミュレーションプログラムにおいて
  - 基本行為の種類を増やす
    本研究では基本行為は５種類に定めていた。基本行為の種類を増やすことで、エージェントはより実世界と近い細かな動きをすることが可能となるので、シミュレーションとしての精度を高めることができる。
  - 障害物を設定するなど、難易度を高める
    実際の川には岩等の様々な障害物があると想定されるため、障害物を想定した行動選択を可能にすることでシミュレーション環境をより実世界に近づけることが課題である。また、川の形状が容易に変更可能となったため、川の形状を変えることで難易度を高めても対応できるプログラムにする必要がある。
- シミュレーションプログラム全体において
  - 実際にカヌーシミュレーション以外のシミュレーションプログラムを作成し、様々なシミュレーション環境下で実験を行う必要がある。

7　謝辞

本論文を作成するにあたり、丁寧かつ熱心なご指導をしてくださった新出尚之准教授に深く感謝の意を表します。また、新出研究室の皆様に感謝いたします。ありがとうございました。

参考文献

[1] 林果穂. 連続的なシミュレーション環境でのエージェントの学習と意思決定の実装について. 2013 年度卒業論文, 奈良女子大学理学部情報科学科, 2014.

[2] 結城浩. 増補改定版 Java 言語で学ぶデザインパターン入門. ソフトバンククリエイティブ, 2004.

[3] 亀村美佳. エージェントの知覚と学習による行動決定の実現. 2013 年度卒業論文, 奈良女子大学理学部情報科学科, 2014.

[4] 宮田玲奈. カヌーレーシングのシミュレーション環境における強化学習について. 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.