



## S<sup>4</sup> Simulation System

### Version 4.3 新機能紹介

#### 特徴

S<sup>4</sup> Simulation Systemは以下のような特徴を持ったシミュレーションシステムです。

- GUIによるモデリング
- psim言語による柔軟なカスタマイズ
- Generatorを用いた柔軟なプロセスモデリング
- 分析機能とグラフ表示機能
- パラメータの最適化機能
- ハイブリッドシミュレーション

#### 新機能

### プラナーフレームワーク

S<sup>4</sup>における同期型エージェントの動作は、step関数に行動ルールを記述する事で実現します。しかし、セルラーオートマトンのような単純なエージェントなら、それほど問題になりませんが、より現実のエージェントに近い行動を記述するのは、困難が生じます。この問題を解決するために、新たにプラナーフレームワークという枠組みを組み込みました。

#### ● 階層型タスクネットワーク

基本的には、どんなエージェントの動作も、IF-THEN-ELSEのルールベースで記述できます。しかしながら、一般的にはエージェントは状態を持ち、ルールには階層があり、複数の行動を正しい順番で実行しなくてはなりません。

これらを一般化したものが、「階層型タスクネットワーク」です。タスクとは、エージェントの行動の単位です。階層型タスクネットワークは、エージェントが目標を持ち、目標を達成するための行動を、条件別にタスクに分解した構造を表したものです。

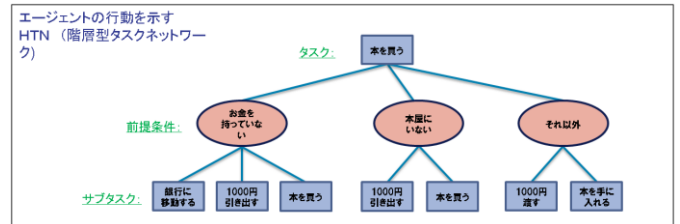


図 1. 階層型タスクネットワークの例

#### ● psim言語と一対一に対応

複雑な行動をするエージェントの実装を容易にするために、階層型タスクネットワークとS<sup>4</sup>のシミュレーション記述言語psim言語コードがほぼ一対一に対応可能になるようなプラナーフレームワークを開発しました。

例えば「本を買う」タスクはエージェントの状態によって、3つに分割されます。さらに、各状態では、複数のサブタスクに分割されます。これをプラナーフレームワークで、実装すると以下ようになります。

```
def 本を買う():
    if お金を持っていない:
        return STATE(銀行に移転する)(), STATE(お金を引き出す(1000), STATE(本を買う)())
    elif 現在地 == 本屋:
        return STATE(お金を引き出す(1000), STATE(本を買う)())
    else:
        return STATE(手渡す(1000), STATE(本を手に入れる)())

def 銀行に移転する():
    if 現在地 == 銀行:
        return None
    else:
        銀行に一歩近付く
        return TICK(), REPEAT()
```

図 2. Pythonコード例

このように、プラナーフレームワークは、階層型タスクネットワークの構造を、そのまま表現です。このフレームワークによって、プログラムの見通しが非常によくなり、効率的にエージェントの行動ルールを記述する事が可能になります。

すなわち、ユーザは階層型タスクネットワークに従って、エージェントの状態を階層的に定義する事だけに集中することが出来ます。

## 強化学習機能の強化

強化学習とは意思決定ルールを最適化する手法の一つで、意思決定者が未知の環境中で逐次的に環境の観測と意思決定（行動）を繰り返していきながら、即時報酬と呼ばれる観測ごとに与えられる報酬値の割引和を最大化するような行動のルールを推定する機械学習手法です。

- 状態 ( $s_t$ ) : エージェントが環境の状態を観測した結果 ( $t$  は時刻)
- 行動 ( $a_t$ ) : エージェントが取った行動
- 即時報酬値 ( $r_t$ ) : 環境の状態などの良し悪しに対応する値

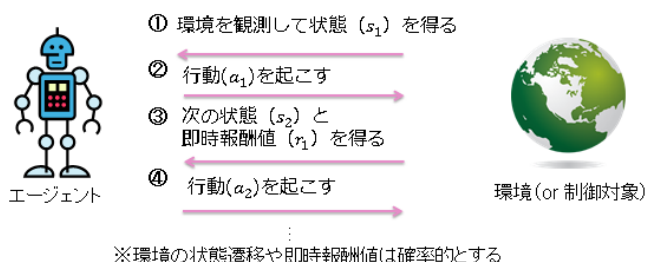


図 2: 強化学習のイメージ

$S^4$  における強化学習では、離散イベントシミュレーションにおけるアイテム(人、モノ、情報)の最適な流れを強化学習によって自動的に学習することが出来ます。また、エージェントシミュレーションにおいても、エージェントの最適な行動を強化学習によって自動的に学習することが出来ます。

Ver4.3 では、この強化学習機能を強化し、よりモデリングがしやすく改良しました。これまでの強化学習機能では、シミュレーション状況の観測や行動の決定と、行動の良し悪しを決めるタイミングが一致している必要がありました。しかし、場合によってはそのタイミングをずらしたい場合があります。

例えば、待ち時間を最小化するようにアイテムの振り分け方を学習させたいとします。このとき、振り分けたアイテムの待ち時間が決まるのは、振り分け先の処理が終わってからであり、次の観測や行動の決定のタイミング(振り分けるタイミング)ではないという問題が生じます。

今回の機能拡張では、行動の良し悪しを決めるタイミングをずらせるような、即時報酬確定アイコンを開発し、アイコンにきた時点で行動の良し悪しを決め

られるようになりました。これにより、より簡単かつ柔軟な強化学習モデリングが可能になりました。

## 社会システムシミュレーション例題集

高橋 真吾 先生(早稲田大学 創造理工学部 教授/社会システムシミュレーション研究所 所長)に監修頂き、社会システムシミュレーションの例題集を作成しています。例題モデルの解説と、 $S^4$  で動作するモデルファイル(Ver4.2以降で動作します)は当社 Web ページで公開しています。社会システムシミュレーションの学習や教育にご利用ください。

現時点までに公開している例題モデルは、以下のものです。

- シェリングの分居モデル
- ライフゲーム
- ゴミ箱モデル
- 繰り返し囚人のジレンマ
- 繰り返し N 人の囚人のジレンマ
- メタ規範ゲーム
- オークション
- 商品の普及モデル
- 情報伝播モデル
- SIR モデル
- 渋滞モデル

例題モデルは今後さらに増やしていく予定です。また、これらの詳しい解説本も高橋先生と作成中です。

## サポート

自社開発製品なので迅速且つきめ細やかなサポートをご提供いたします。また、お客様の目的に合わせたカスタマイズや周辺ソフトウェアの開発にも積極的に対応いたします。弊社の長年にわたるシミュレーション・データ解析分野の経験で蓄積したノウハウを元に、お客様の問題に最適なソリューションを提供いたします。

お問い合わせ

**NTT DATA**

株式会社NTTデータ 数理システム

〒160-0016

東京都新宿区信濃町 35 番地 信濃町煉瓦館 1 階

TEL : 03-3358-6681 FAX : 03-3358-1727

E-mail [s4-info@msi.co.jp](mailto:s4-info@msi.co.jp)

URL <http://www.msi.co.jp/s4/>