

S³ Simulation System 新機能紹介

株式会社 数理システム 雪島正敏

s3-info@msi.co.jp

1. 概要

戦略立案・収益予測・リスク分析など、現実の世界の問題には多くの不確定要素が含まれる上に、システムが複雑で予測が困難な場合が少なくありません。しかし、実際のシステムで試すにはコストも時間も掛かり、また、まだシステムが出来ていない場合なども考えられます。このような状況で威力を発揮するのがシミュレーションです。シミュレーションとは、現実のシステムをモデル化し、その振る舞いを分析・予測する問題解決手法です。現実世界の問題をコンピュータ上に再現してシミュレートすることにより、実際のシステムを変更することなく、様々な条件におけるシステムの挙動を調べることが出来ます。

S³ Simulation System(以下 S-cube)は、株式会社 数理システムにより独自に開発された離散イベントシミュレーションツールです。離散イベントシミュレーションとは、システムの状態変化を起こす事象が離散的に起こるシステムを対象としたシミュレーションです。工場などの生産システム、サプライチェーンなどの流通システム、銀行の窓口や ATM、通信システム、コールセンターなど、確率的な振る舞いをする、待ち行列型の様々な領域のシミュレーションを行なうことが出来ます。また、どこがボトルネックになっているか、どこで無駄なコストを発生しているかなどの分析を行うことが出来ます。S-cube はこれらのシミュレーションが簡単に出来る操作性と、他に類をみない自由度を併せ持つシミュレーションシステムです。本紹介では S-cube の最新版(Ver.2.0)に含まれる機能について紹介します。

2. 特徴

S-cube は離散イベントシミュレーションツールです。離散イベントシミュレーションとは、システムの状態変化を起こす事象が離散的に起こるシステムを対象としたシミュレーションで、所謂、待ち行列型の現象を分析・評価します。例えば、ジョブ・シヨップとはジョブにより機械の処理順序が異なる多数のジョブを扱う生産形態ですが、このモデルにおいては、ジョブの到着事象が離散的に発生します。それぞれのジョブに従い機械で処理されますが、機械の数は有限である為にここに「待ち」が発生します。この待ちにより、実際に処理される時間がさまざまに変化します。ジョブの到着間隔や、機械の数、処理時間を設定し、シミュレーションを走らせることで、それぞ

れの機械での待ちの発生状況や製品のスループットなどを分析します。

S-cube の特徴を挙げると以下ようになります。

- 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
- 入出力結果のサマリ
- 分析機能とグラフ表示機能 ※ Ver.2.0 で拡張された機能
- パラメータの最適化機能 ※ Ver.2.0 での新機能

以下にそれらについて紹介いたします。

【GUI によるモデリング】

シミュレーションに表れる基本部品をマウス操作によりモデル編集パネルに配置し、それらを線で結ぶことで簡単にシミュレーションモデルを作成出来ます。各部品は様々なパラメータを持ち、それらを設定することで個々の部品の動作を決定します。

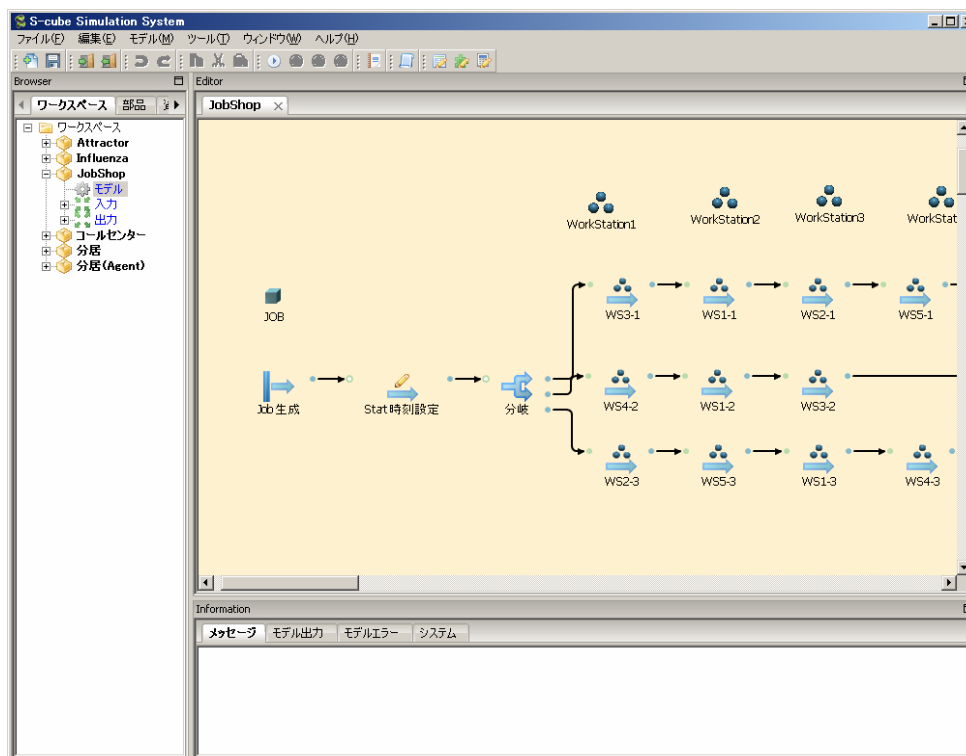


図 S-cube 画面

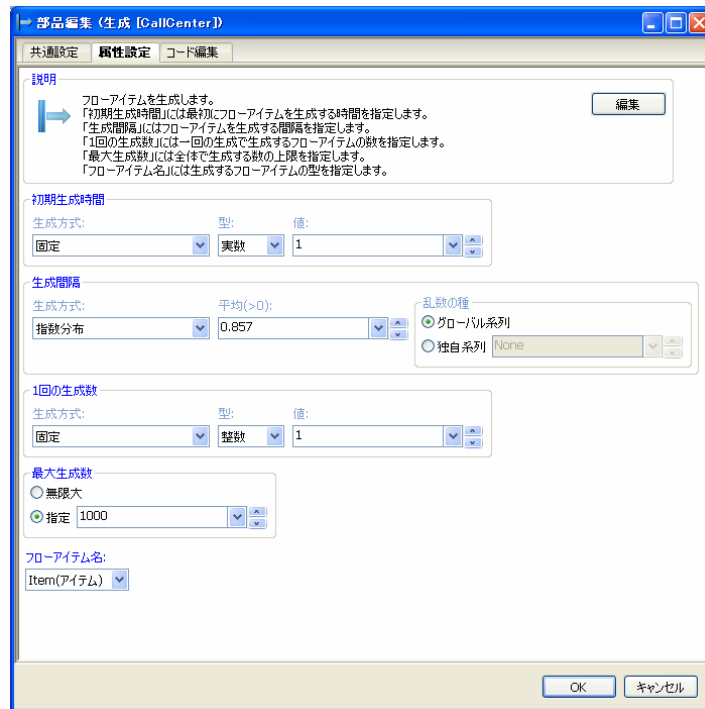


図 パラメータ設定画面

シミュレーションモデルとその入出力をまとめたものをプロジェクトとして管理します。プロジェクトは同時に複数編集可能で、作成したモデルまたはその一部は、別なプロジェクトへコピーし利用することも可能です。

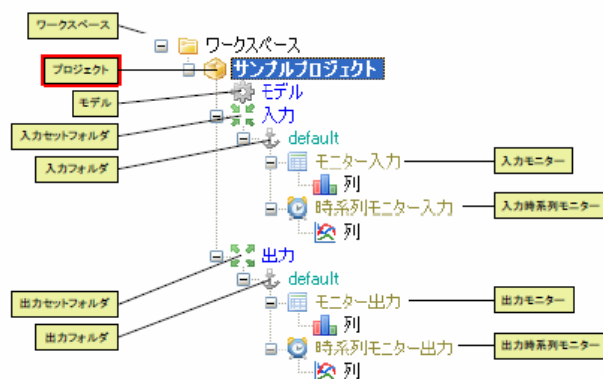


図 ワークスペースタブ

【psim 言語による柔軟なカスタマイズ】

全ての部品は psim 言語とよばれる Python 言語上に実装されたシミュレーション言語を用いて動作が記述されています。パラメータ設定による変更以上に、部品の動作をより柔軟に変更したい場合には、これらを直接、修正することでより柔軟に部品の動

作を設定することが可能となります。

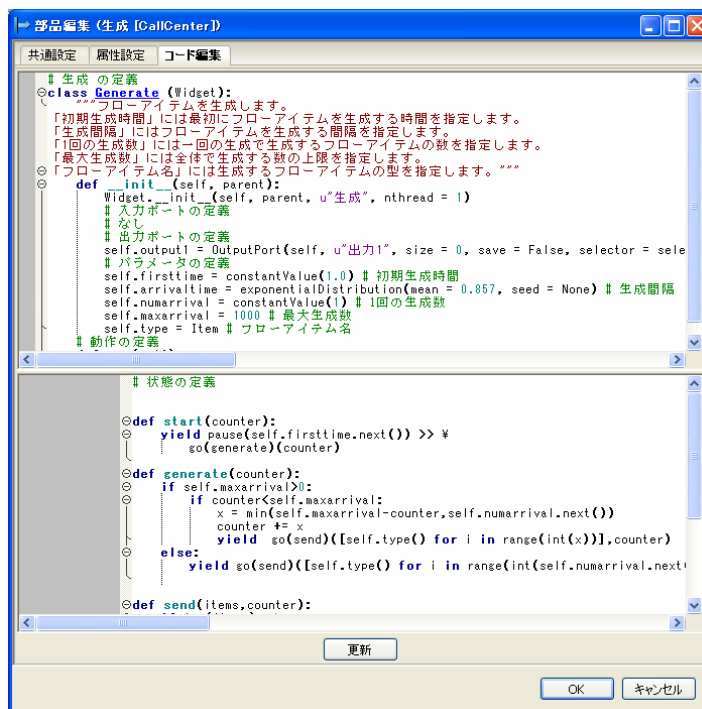


図 コード編集画面

また、ユーザ独自の分析スクリプトやカスタムコードを定義することが可能で、Pythonの標準ライブラリなどを `import` し、独自の理論分布や経験分布に従う乱数発生器を定義したり、独自の統計処理関数を定義し、シミュレーション実行中に呼び出すことが可能です。

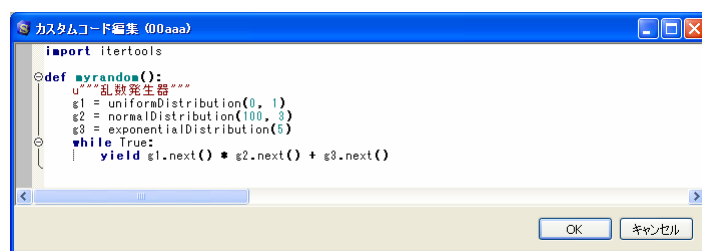


図 カスタムコード編集画面

更に、GUIで作成されたモデルは全て `psim` 言語に変換されてから実行されます。`psim` 言語を理解していなくても GUIによるモデリングやシミュレーションを行なうことが出来ますが、`psim` 言語を取り扱うことで、より複雑で柔軟なモデルを記述することが可能となります。

【Generator を用いた柔軟なプロセスモデリング】

psim 言語では、Python の Generator 機能を用いた高速なコンテキストスイッチを行なうスケジューラが搭載されているため、大量のプロセスを同時に実行する事が出来ます。資源(Facility、Store、Tank、Event)を利用する待ち受け式(ガード式)が各種用意されております。また、それらの複合待ち受けとして、And 待ち受け、Or 待ち受け、逐次待ち受けも可能です。And 待ち受けは複数のガード式が全て成立するのを待ち受けます。Or 待ち受けでは複数のガード式のうち一つでも成立するのを待ち受けます。逐次待ち受けでは順序を指定して複数のガード式を待ち受けます。これらを用いることで、資源の取得待ち受けを行なうがある時間経過したら諦めることや、資源の取得を待ち受け、取得後ある時間経過するまで待つこと等の記述を柔軟に行なうことが出来ます。また、現在実行中のプロセスを別なプロセスに置き換えるガード式を書くことも出来ます。これにより状態遷移を簡単に書くことも可能です。

【出力結果の分析】

各部品で観測された結果から様々な基礎統計量を得ることが出来ます。また、データを表示したり、ヒストグラムを作成したり、時系列グラフを作成する事が出来ます。

名前	観測数	平均	標準偏差	95%信...	95%信...	変動係数	最小値	25%値	中央値	75%値	最大値
-default											
タンク											
<input type="checkbox"/> 入力待ち要求	0	---	---	---	---	---	---	---	---	---	---
<input type="checkbox"/> 入力待ち合計	0	---	---	---	---	---	---	---	---	---	---
<input type="checkbox"/> 出力待ち要求	0	---	---	---	---	---	---	---	---	---	---
<input type="checkbox"/> 出力待ち合計	0	---	---	---	---	---	---	---	---	---	---
<input type="checkbox"/> バッファ	0	---	---	---	---	---	---	---	---	---	---
ファシリテイ											
<input type="checkbox"/> 入力待ち要求	773	0	0	0	0	---	0	0	0	0	0
<input type="checkbox"/> バッファ	773	0.526162	0.500037	0.490857	0.561468	0.950347	0	0	1	1	1
ファシリテイ利用-入力1											
<input type="checkbox"/> 入力待ち要求	1031	0.473838	0.499944	0.443285	0.50439	1.0551	0	0	0	1	1
<input type="checkbox"/> 入力待ち合計	1031	0.473838	0.499944	0.443285	0.50439	1.0551	0	0	0	1	1
<input type="checkbox"/> 出力待ち要求	1031	0	0	0	0	---	0	0	0	0	0
<input type="checkbox"/> 出力待ち合計	1031	0	0	0	0	---	0	0	0	0	0
<input type="checkbox"/> バッファ	1031	0.913344	2.3856	0.767554	1.05913	2.61194	0	0	0	1	15
ファシリテイ利用-待ち時間											
<input type="checkbox"/> 待ち時間	516	1.76577	4.04198	0.490857	0.561468	2.28907	0	0	0.0023364	1.41846	22.0845
ファシリテイ利用-通過時間											
<input type="checkbox"/> 通過時間	516	2.783	4.27539	2.41324	3.15276	1.53625	0.00057980	0.507394	1.35598	2.78343	23.7337

図 サマリ表示画面

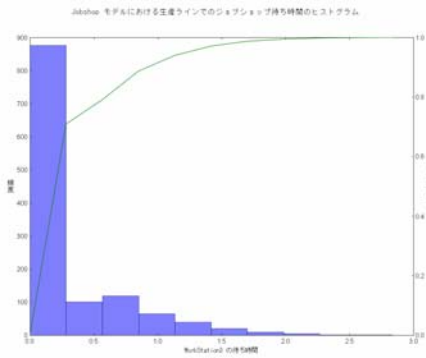


図 ヒストグラム表示画面

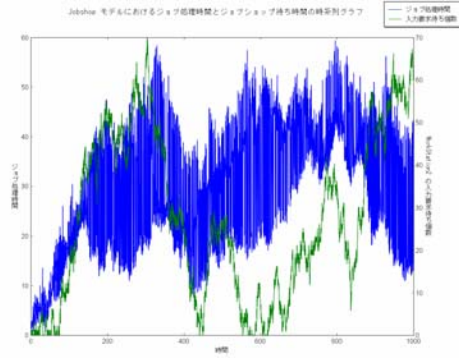


図 時系列グラフ表示画面

3. 分析機能とグラフ表示機能

Ver.2.0 から強化された機能の一つに分析機能とグラフ表示機能があります。分析機能では、与えられたデータの分布を推定する機能、分布の適合度を検定する機能、時系列データの振る舞いを推定する機能が追加されました。

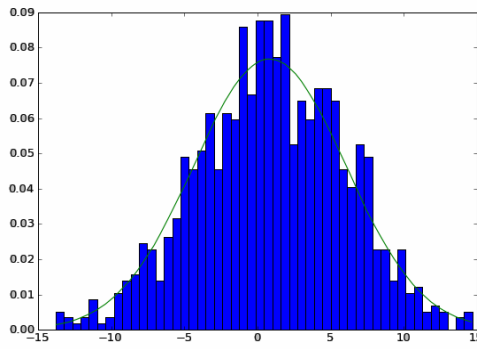


図 分布推定の様子

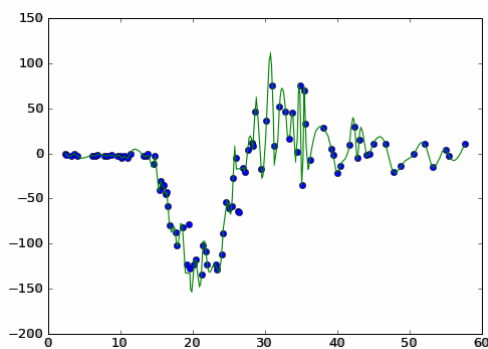


図 時系列推定の様子

グラフ表示機能では、ヒストグラムや時系列プロットのほかにも、散布図や棒グラフ、折れ線グラフ、円グラフなど用途に応じた様々なグラフ表示を行えます。

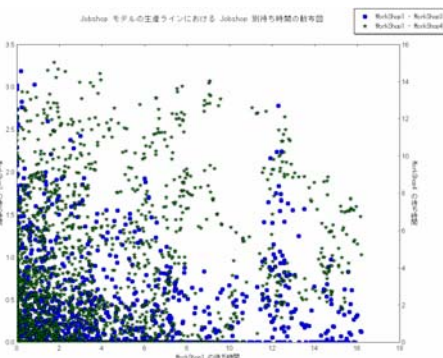


図 散布図表示画面

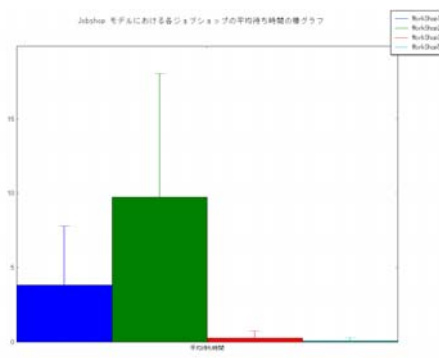


図 棒グラフ表示例

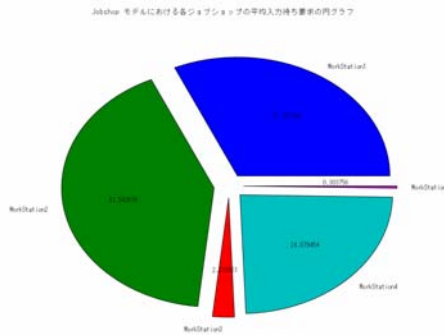


図 円グラフ表示画面

またシミュレーション実行中に、各部品で観測している値をリアルタイムに表示することもできるようになりました。これを用いることにより、実行時の過渡的な様子をリアルタイムに観察することが可能となり、膨大な実行時間を必要とするシミュレーションであっても、途中経過を用いてすぐに分析を始めることが出来、シミュレーションと分析を効率的に行えます。

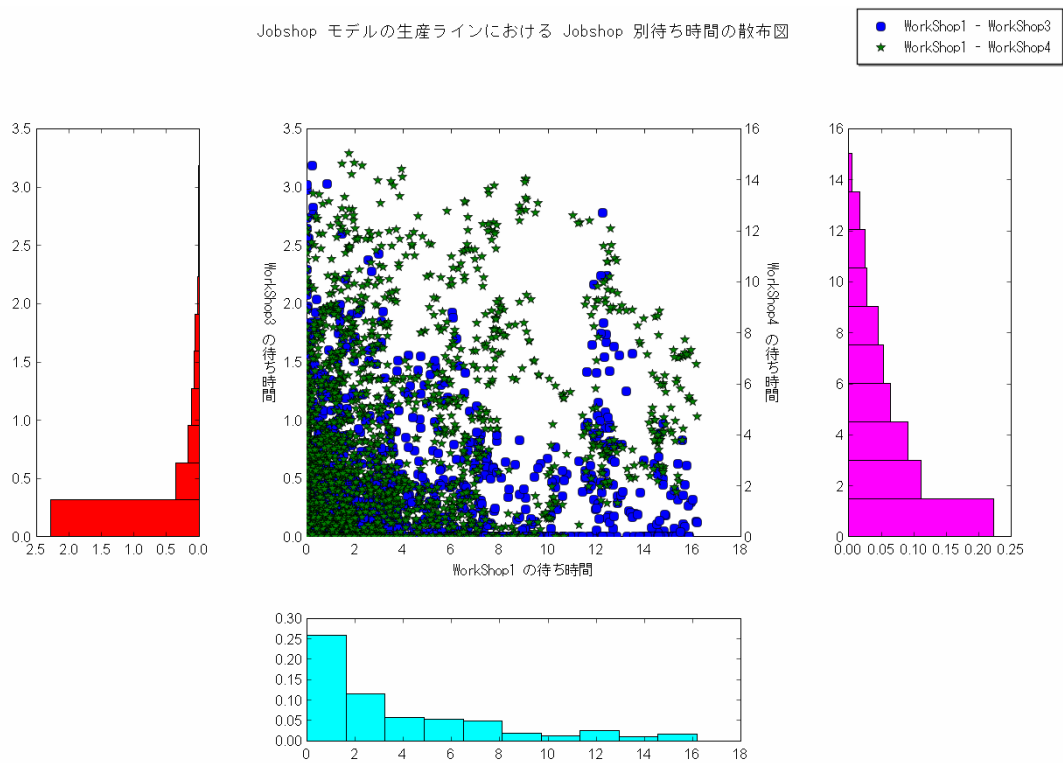


図 散布図とヒストグラムの組み合わせ例

これらのグラフは、ユーザが GUI を用いて各項目をグラフィカルに編集することが可

能です。さらにそれだけに留まらず複数のグラフを一枚のシートに自由に配置したりなど、思いのままにグラフを作成できます。一度作成したグラフはデータを変更することで簡単に使い回すこともでき、面倒なグラフ作成のフェイズを強力にサポートします。

4. 最適化

シミュレーションでは、様々な条件を変えながらシステムの振る舞いを分析・評価します。これらの評価は、通常「what if」形式で行なわれます。しかし、この方法では、余り多くの可能性を探索することが出来ませんし、また、得られた結果が最適である保証はありません。そこで、S-cube では、パラメータの最適化が行なえる機能を追加しました。

シミュレーションの最適化の難しい点は、一つは、通常最適化問題と異なり、シミュレーションの結果は確率的に変動するために、厳密なシステムの評価(目的関数の評価)が行えないということです。もう一つは、システムの評価をする為に必要な1回のシミュレーションに多くの時間が掛かることです。システムの挙動が確率的であるために、それを評価する際には単純にシミュレーションの結果を用いることが出来ません。その為、期待値を取るなど何かしらの推定を行なう必要があります。しかし、システムの評価に時間が掛かるために、システムを評価するために余りに多くの試行を行なうわけには行きません。また、システムの評価に時間が掛かるために、なるべくシステムの評価(目的関数の評価)を行なう回数を減らした最適化手法が求められます。S-cube では NUOPT のアドオン機能として提供されている DFO(Derivative Free Optimization)と呼ばれる手法を用いた最適化エンジンを用いてパラメータの最適化を行ないます。

5. まとめ

S-cube は部品を配置することで簡単にモデルを構築しシミュレーションを行なうことが出来る離散イベントシミュレーションツールです。また、簡単な操作性のみならず、独自の psim 言語により、他に類を見ない非常に高い自由度をあわせ持ちます。Ver.2.0からはグラフ機能も強化され、実行時にリアルタイムに結果を確認することも出来るようになりました。また、最適化機能も搭載され、システムの最適なパラメータを求めることもできるようになりました。今後は、連続系や Agent シミュレーションなど、離散イベント以外のモデリング手法や、製造業や通信業、流通業に特化したモジュール・テンプレートの開発を予定しています。