S4 Simulation System 新機能紹介
～エージェントシミュレーション～

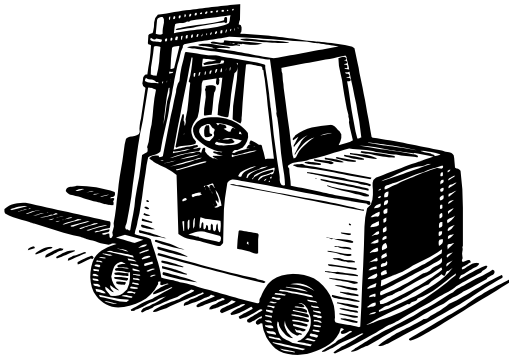
株式会社NTTデータ数理システム

Global IT Innovator NTT DATA Group

NTT DATA MATHEMATICAL SYSTEMS INC.

内容

- 概要
- シミュレーションとは
- エージェントシミュレーションとは
- S4 Simulation System の紹介
- エージェントシミュレーションの応用
- エージェントシミュレーションの事例
- まとめ



概要

- 本報告書では、S4 Simulation Systemの新機能を紹介します。
- 今回追加された機能は、エージェントシミュレーションを記述するためのフレームワークです。
- エージェントシミュレーションのご紹介と、その応用例を解説します。
- 最後に、ITS世界会議2013　にて発表された事例をご紹介します。

- 名称変更について
  - 次期リリースより、S^3 Simulation System　から S^4 Simulation System に改名されます。
  - S^4 はエスクワトロ（S-Quattro）と読みます。

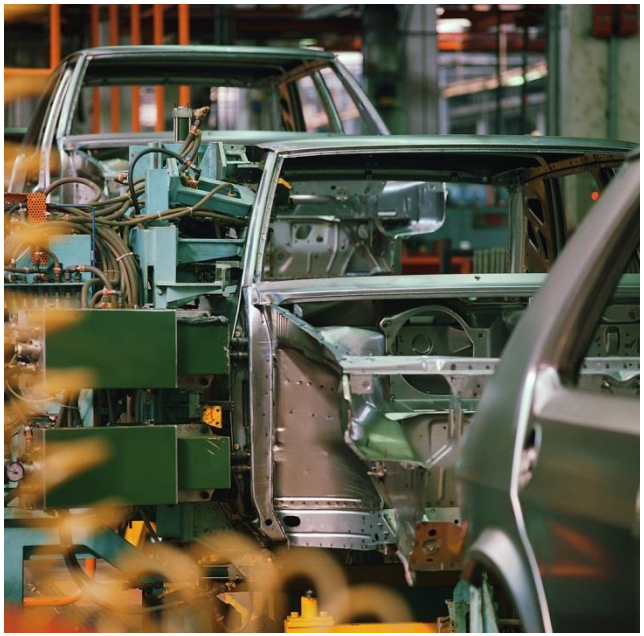
ユーザーコンファレンス2013

シミュレーションとは

シミュレーション概要

現実のシステムをモデル化（模擬）し、
その振る舞いを分析・予測する問題解決手法

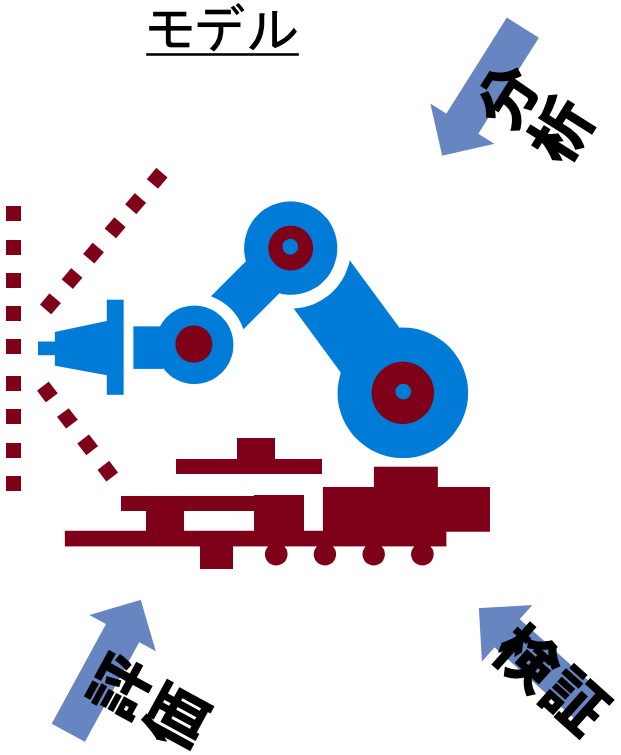
現実の複雑なシステム



目的により
特徴を抽出し
簡略化

モデル化

モデル



シミュレーションのメリット

- 実際のシステムで試すには時間・費用が掛かる
- 実際のシステムが存在しない
- 問題が複雑で解けない
- 複数の解を比較したい

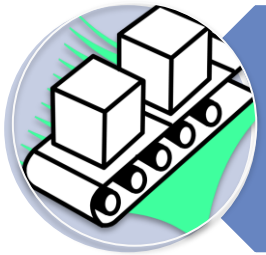
- **挙動の確認**

  モデルを調べることで実際のシステムの挙動や、様々な量を（条件を変えながら）分析可能

- **不確定要素の影響の調査**

  確率的に変動する要因を持ったシミュレーションを行ない、その結果を統計的に分析し問題解決の糸口に

シミュレーションの種類



離散イベントシミュレーション

・状態変化が離散的に発生するような現象をシミュレートする
・排他的なサービスを利用するために発生する待ち行列の時間変化シミュレーションなど

連続型シミュレーション（システムダイナミクス）

・状態量が連続的に変化するような現象をシミュレートする
・微分方程式であらわされたモデルにおける各状態量の時間変化シミュレーションなど

エージェントシミュレーション

・全体の挙動をエージェントの挙動からシミュレートする
・一定のルールに従い自律的に行動するエージェントが、相互に作用し合いながら行動する事によって生じる現象のシミュレーションなど

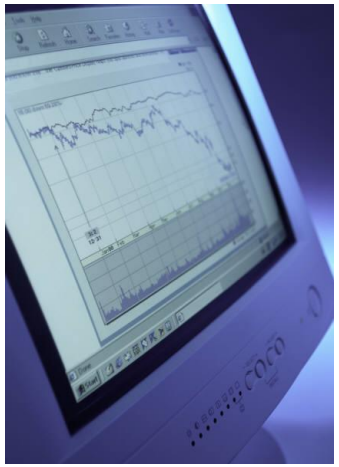
本発表では新機能であるエージェントシミュレーションを中心に説明します

ユーザーコンファレンス2013

エージェントシミュレーションとは

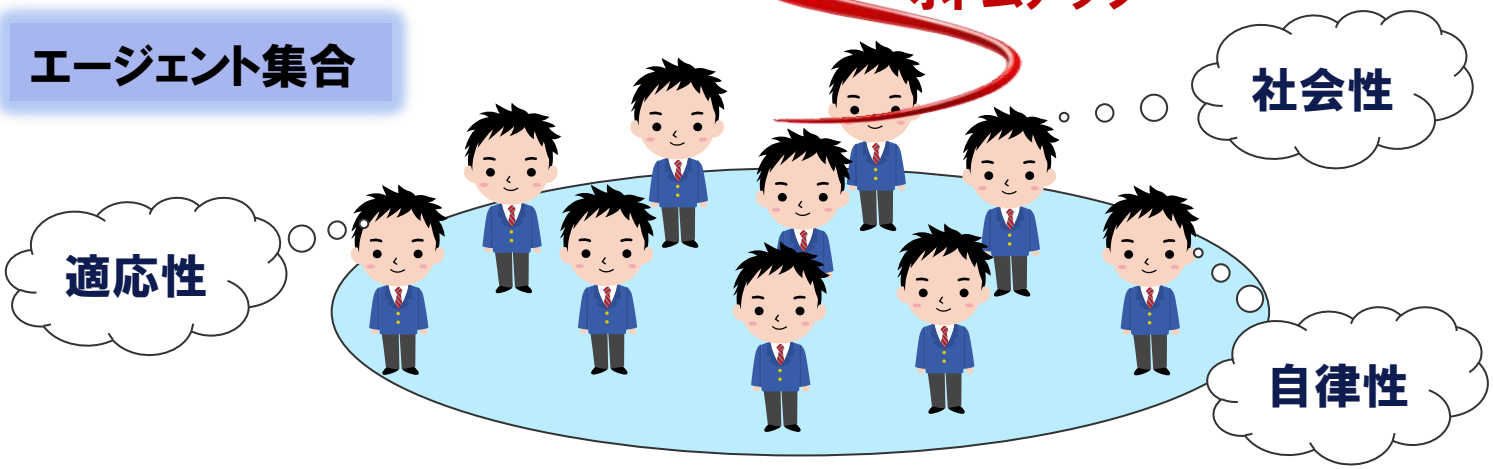
エージェントシミュレーション概要

エージェントシステム



ボトムアップ

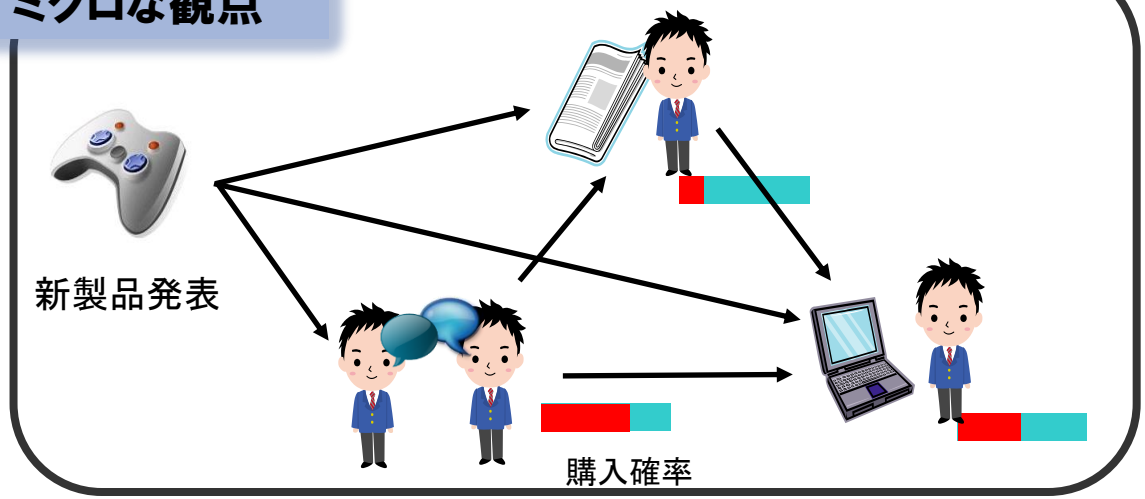
エージェント集合



エージェント=自己の行動を自律的に決定する主体

ミクロ/マクロ的な観点

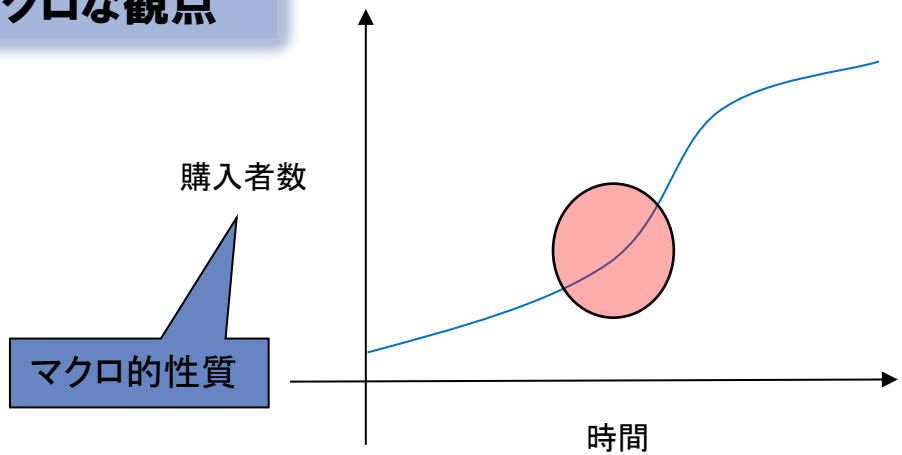
ミクロな観点



エージェントベースのシミュレーションでは、エージェント間のつながりや、エージェントの状態や行動に対して、ある程度複雑な制約を課すことができ、より現実世界に近いモデル化を行うことが可能となる。

ミクロ的なエージェントの振る舞いにより、**マクロ的な性質**が変化する。
このように、色々な要因が絡み、エージェント同士が互いに影響した結果を「**創発**」と呼ぶ。

マクロな観点



エージェントベースモデリング

エージェントシミュレーションをモデル化する際に必要になる観点

観点	特徴
同期/非同期	エージェントの状態の変化するタイミング 同期ではすべて同じ、非同期ではエージェントごとに異なるタイミングで変化する(後述)
エージェント集合	個々のエージェントの状態だけではなく、エージェント全体としての状態をどのように考えるか
環境	エージェントが振舞う環境 交通シミュレーションであれば道路ネットワークなど
可視化	環境の可視化

今考えているモデルがどのようなモデルかを分析し最適なモデル化を選択する

同期/非同期エージェント

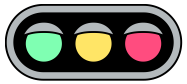
- 同期エージェント
  - 全てのエージェントの状態が固定タイミングで変化する
  - 例えば、交通シミュレーション、物理シミュレーションなど

メリット	エージェント数の増加に対してスケールしやすい。（内部的なスケジューリングが簡単であるため）
デメリット	より精度の高いシミュレーションにするには、すべてのエージェントのステップ幅を短くする必要がある。また、（間隔の分散の大きい）離散イベントの処理は、効率が悪い。

同期エージェントの例

x_{i-1}, v_{i-1}, a_{i-1}

x_i, v_i, a_i



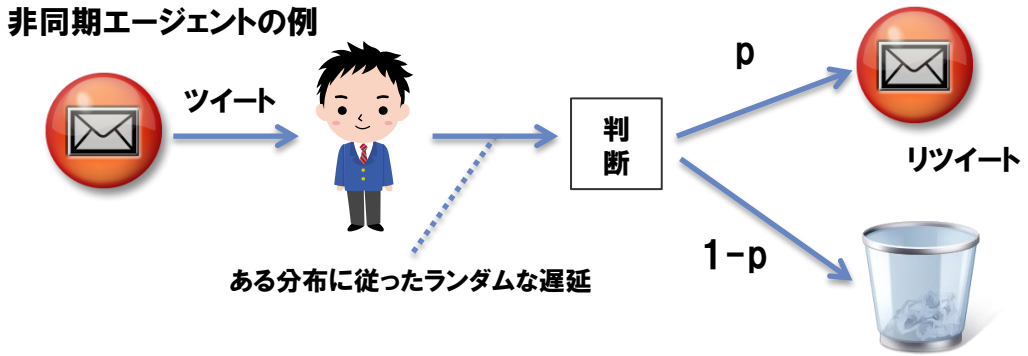
C_k: 信号機の状態

各車両は前車の位置、速度、加速度、信号機の状態などから、Δt 後の加減速を決定する

同期/非同期エージェント

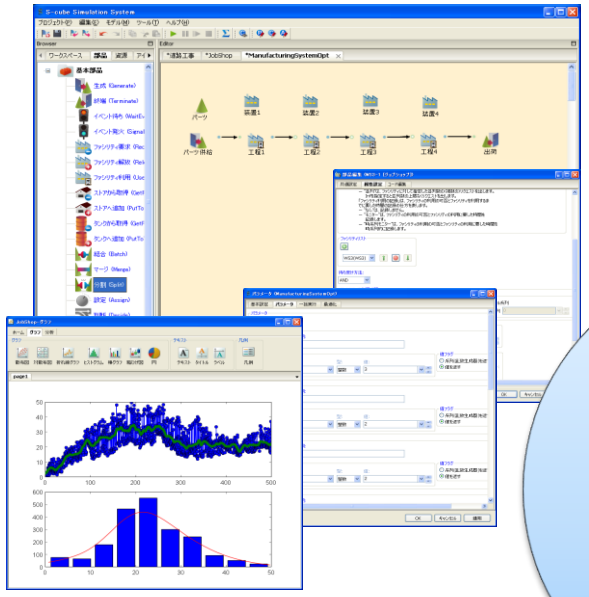
- 非同期エージェント
  - 個々のエージェントの状態が独立に離散タイミングで変化する
  - 例えば、ツイッターの伝播シミュレーション、購買行動シミュレーションなど

メリット	イベント発生間隔が疎なら、高速な離散イベントシミュレーションができる
デメリット	エージェント数の増加に対して、あまりスケールしない。

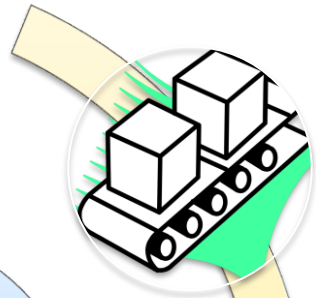


ユーザーコンファレンス2013

S4 Simulation System の紹介



S4 Simulation System



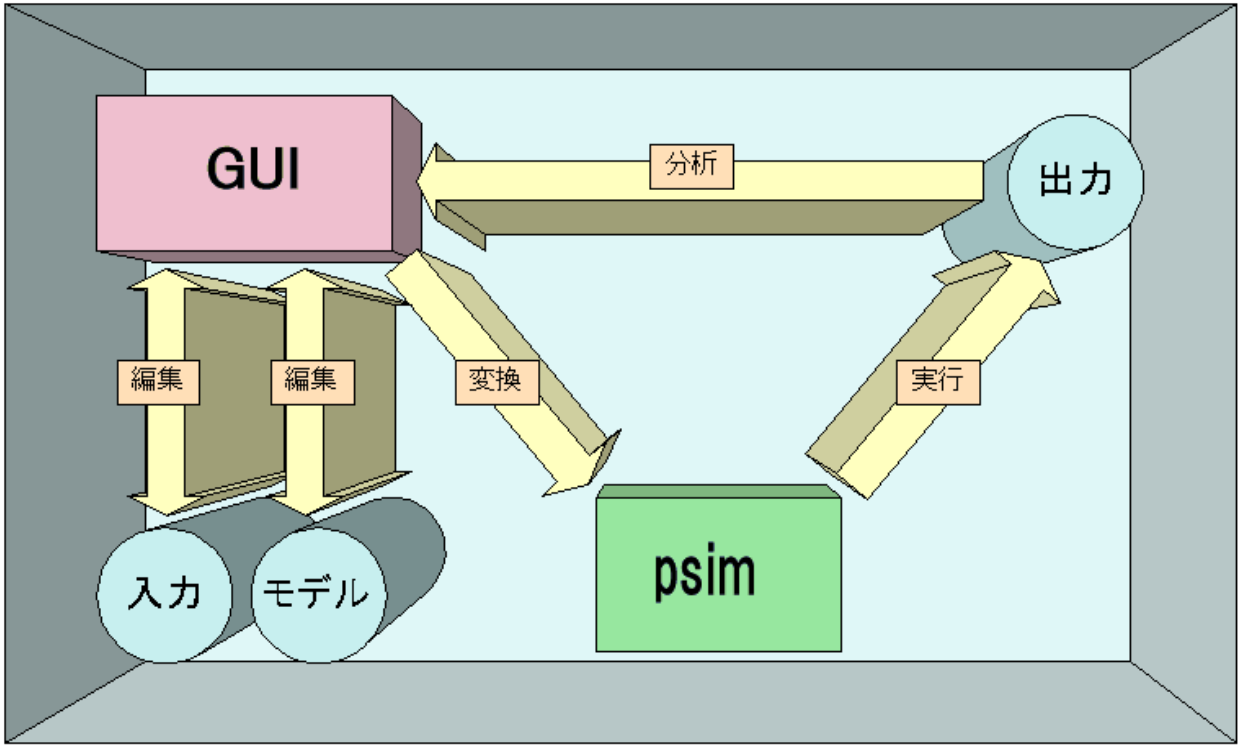
離散イベントシミュレーション

連続シミュレーション

エージェントシミュレーション

- GUIによる直観的なモデリング
- 柔軟なカスタマイズ性能
- ハイブリッドシミュレーション
- グラフ・統計分析
- 最適化・感度分析・実験計画

システム構成



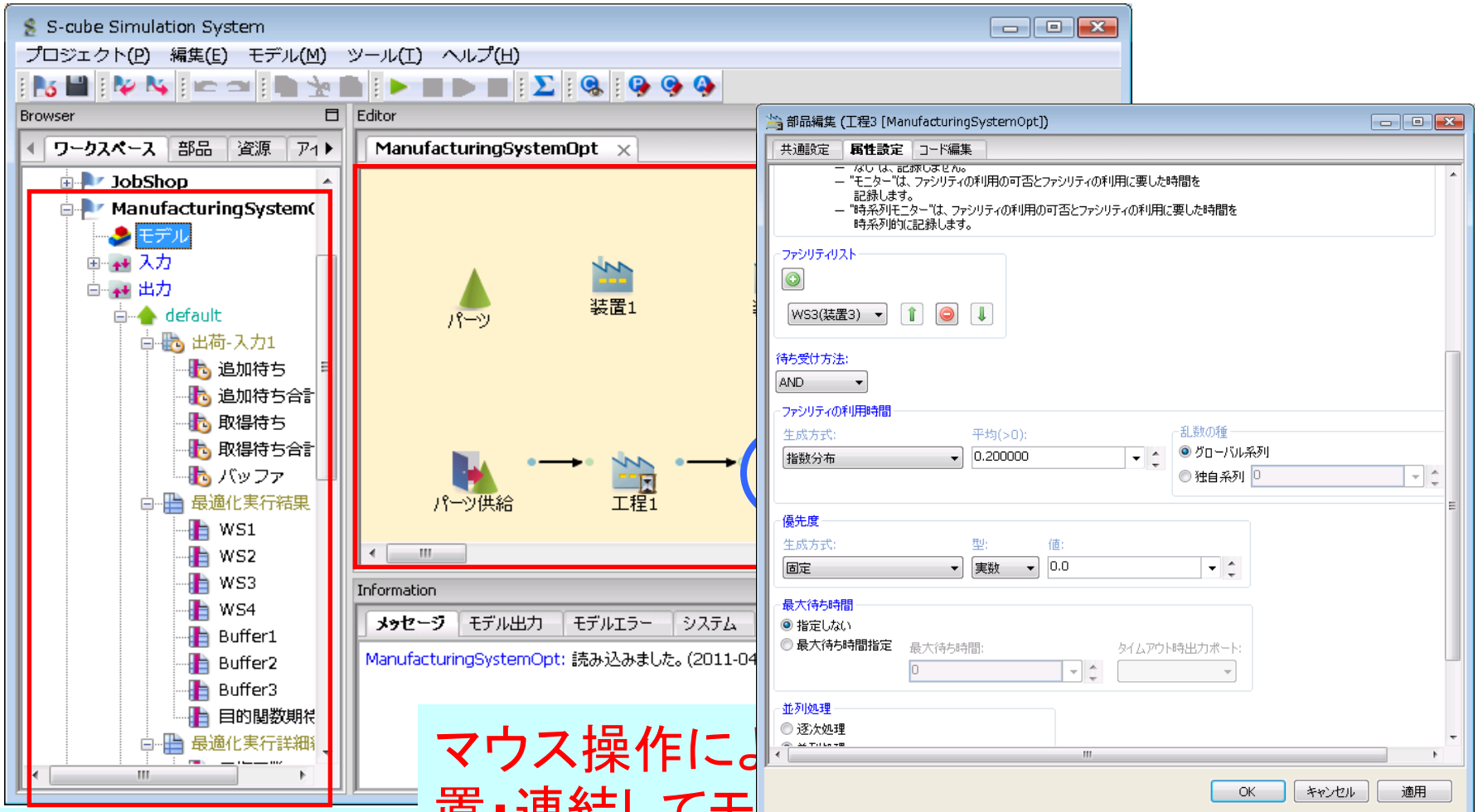
GUI

- ☆ wxPythonを用いて開発
- ☆ 入出力データの管理
- ☆ モデルの管理・編集
- ☆ 実行

psim

- ☆ Python言語上で動作するプロセス指向のシミュレーション記述言語
- ☆ イベント処理エンジン
- ☆ 乱数生成や分布推定などの分析機能も併せ持つPythonライブラリ集

GUI



マウス操作によ
置・連結してモデル作成

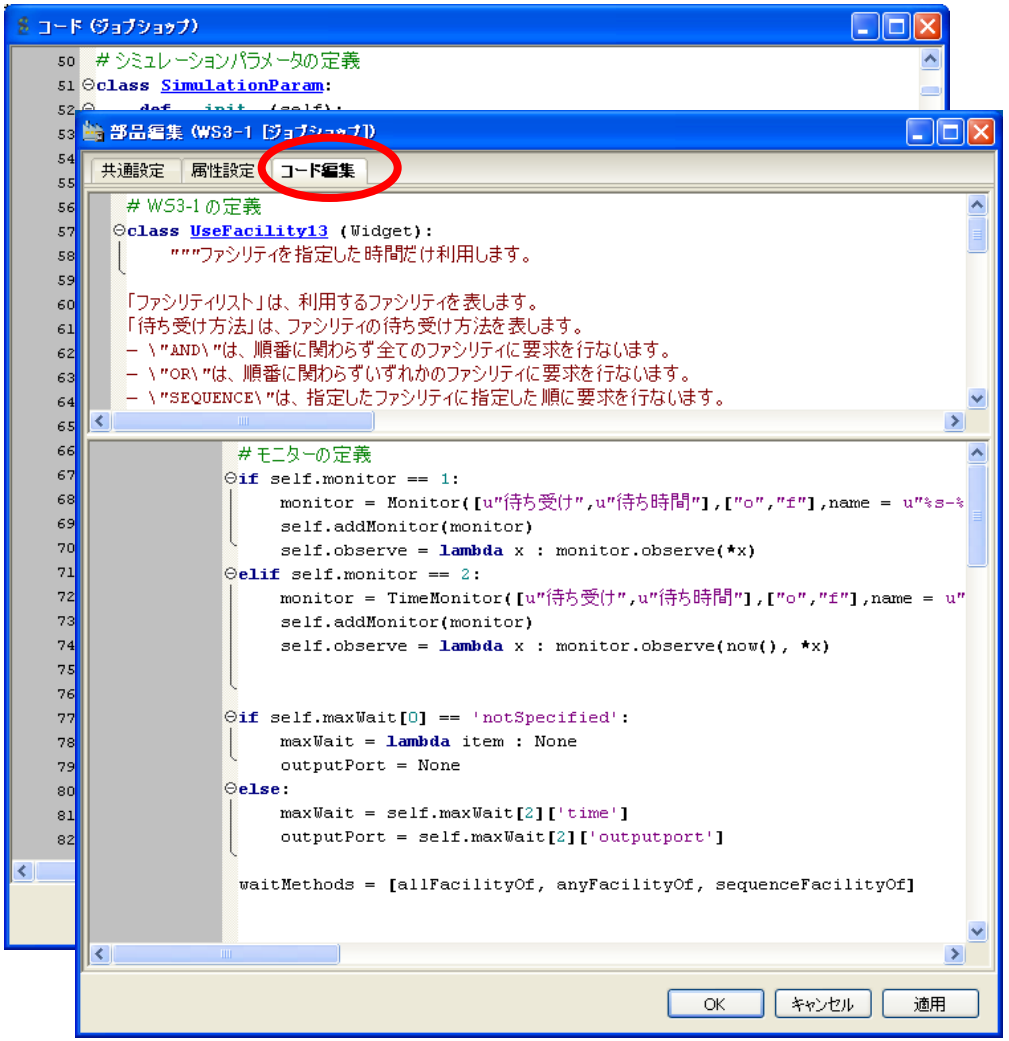
モデルと入出力をあわせて
プロジェクトとして管理

部品のパラメータ設定画面

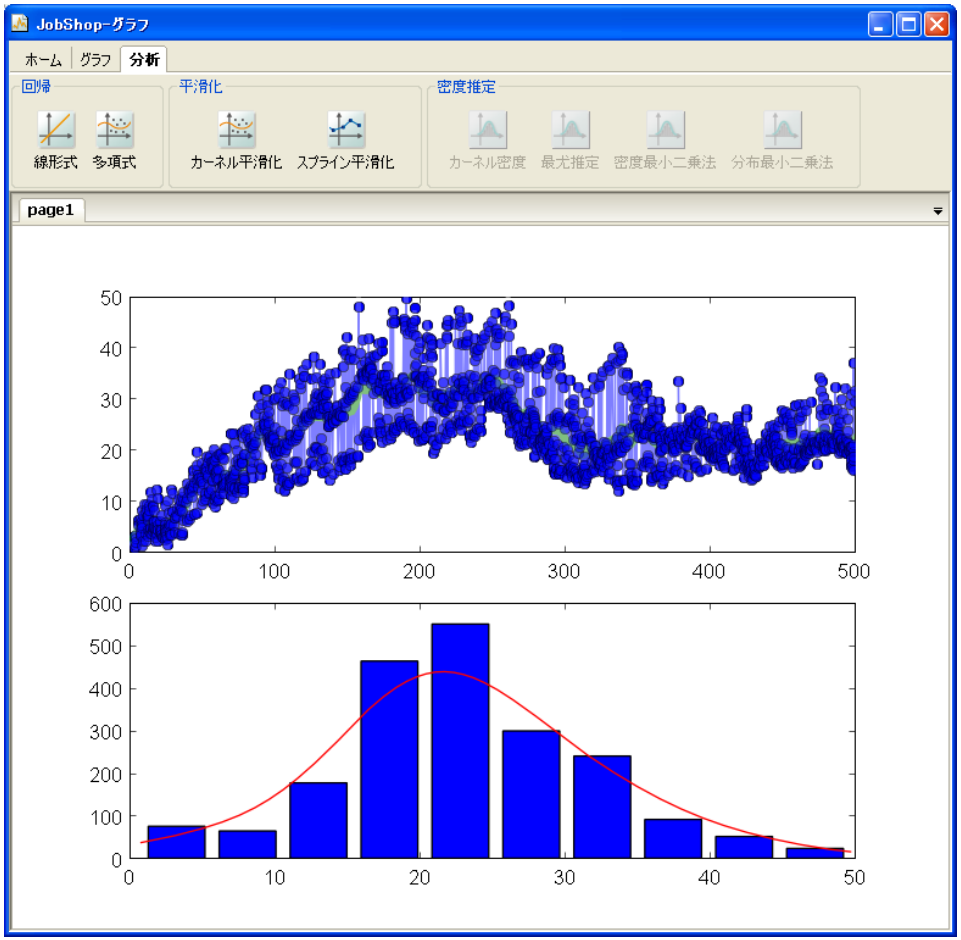
psim言語

- 作成したシミュレーションモデルはpsim言語に変換され実行される。
- psim言語はPythonのライブラリである。

自らコードを書かなくても実行可能であるが、コードを編集することで**より柔軟なモデルの記述**が可能に



グラフ機能・分析機能



- **グラフ種類**
  - 散布図
  - 対散布図
  - 折れ線グラフ
  - ヒストグラム
  - 棒グラフ
  - 箱ひげ図
  - 円グラフ

- **分析機能**
  - 回帰
    - 線形
    - 多項式
  - 平滑化
    - カーネル平滑化
    - スプライン平滑化
  - 密度推定
    - カーネル密度推定
    - 最尤推定
    - 密度最小二乗法
    - 分布最小二乗法

機能

GUI
- GUIを用いてモデルを作成・編集する機能
- モデル、入力データ、出力（結果）データをまとめてプロジェクトとして管理
- モデル全体のパラメータの管理
- 実行モードの管理

プロセス管理
- コルーチン（PythonのGenerator機能）を用いた軽量プロセス管理
- サブプロセスや状態遷移

グラフ表示
- 入力・出力データのグラフ表示
- 実行時のリアルタイムグラフ表示

統計解析
- 平均・分散などの統計量算出機能
- 入力・出力データの分布推定機能
- 入力・出力データの補間・平滑化機能
- 入力・出力データの回帰機能

乱数生成
- 乱数の生成機能

最適化
- DFOを用いた最適化機能
- PSOを用いた最適化機能
- PSOを用いた多目的最適化機能

感度分析
- シミュレーションパラメータに対する感度分析機能
- トルネードグラフ、スパイダーチャートによる結果表示

実験計画
- D-最適計画作成・シミュレーション実行機能

並列実行
- 複数のパラメータに対するシミュレーションの並列実行機能

外部連携
- Visual Mining Studioとの連携機能

連続型シミュレーション
- ODEソルバー(Runge-Kutta 法, BDF 法, Adams 法)

エージェントシミュレーション
- 同期/非同期エージェント、環境（グラフ、連続空間、離散空間）、可視化機能

指数分布
正規分布
対数正規分布
一様分布
ベータ分布
ガンマ分布
アーラン分布
パレート分布
ワイブル分布
カイ2乗分布
F分布
ロジスティク分布
非心カイ2乗分布
非心F分布
コーシー分布
t分布
三角分布

二項分布
幾何分布
超幾何分布
負の二項分布
ポアソン分布

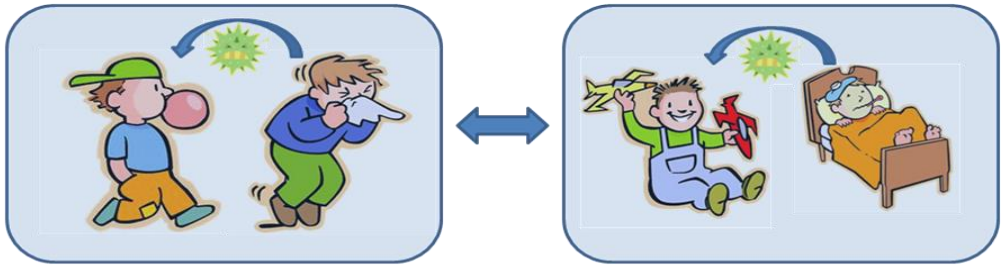
経験分布
再生
ステップ

ユーザーコンファレンス2013

エージェントシミュレーションの応用

感染シミュレーション

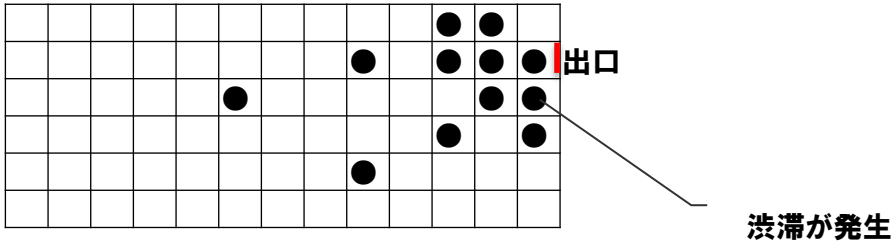
- インフルエンザなどの感染モデルには、SIRモデルがよく利用される。
- SIRモデルは、非感染者が感染者に接触すると一定の確率で感染し、感染者は一定の確率で回復し、免疫保持者になるというモデルである。
- SIRモデルは、システムダイナミクスとしてモデル化される事が多いが、その場合はあくまでもマクロ的に感染の広がりをシミュレーションしたものであり、ミクロ的な人と人のつながりを考慮したシミュレーションを行えない。
- 例えば、エージェントシミュレーションなら、「人はそれぞれ、所属する学校（職場）と自宅があり、それらを往復する」というモデルも簡単に表現できる。



- 他にも、個々の感染率の違い、個々の行動の違い（マスクをするなど）、学級閉鎖などのイベントも容易に取り込むことができる。

避難行動シミュレーション

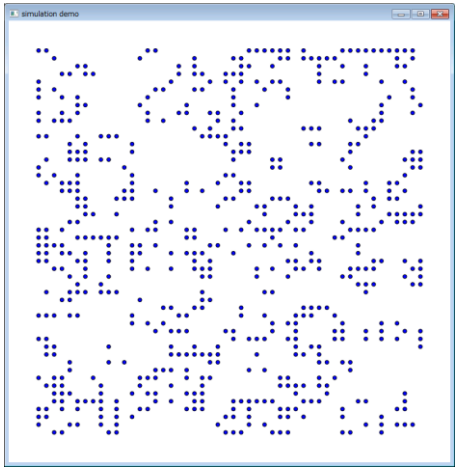
- 個々の避難対象のエージェントは、避難行動を行うが、例えば避難経路への出口が狭ければ渋滞が発生する。これはひとつのセルには高々ひとつのエージェントしか属すことができないという制約を課す事で、シミュレート可能である。



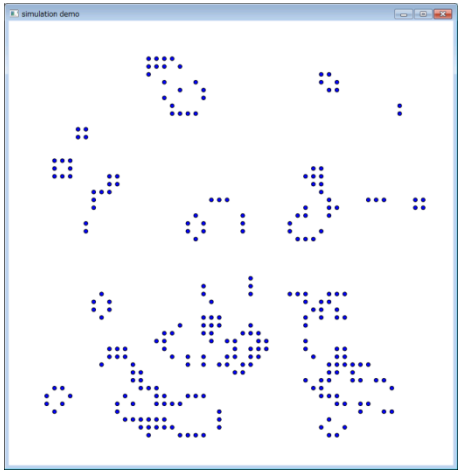
- このようなセルを使ったシミュレーションは、セルオートマトンと呼ばれる。シミュレーション空間上に多数のセル(細胞)が配置されており、それぞれのセルには近傍がある。各セルは状態を持っているが、各セルの次の状態は近傍のセルの状態によって決定する、というモデルである。
- 他に、交通シミュレーション、様々な施設内の動線(人の流れ)シミュレーションなどにも応用される。

ライフゲームエージェント

- ライフゲームも、単純なルールから複雑な結果が得られるセルオートマトンの1例である。
- 8 方格子上で、各セルは以下のルールで「生」と「死」の状態を遷移する。
  - 自分の状態が「生」の場合
    - 回りに生きたセルが 1 以下の場合、死滅する。(過疎)
    - 回りに生きたセルが 2 または 3 の場合、生存する。
    - 回りに生きたセルが 4 以上の場合、死滅する。(過密)
  - 自分の状態が「死」の場合
    - 回りに生きたセルが、3 の場合、誕生する。
    - それ以外の場合、変化しない。



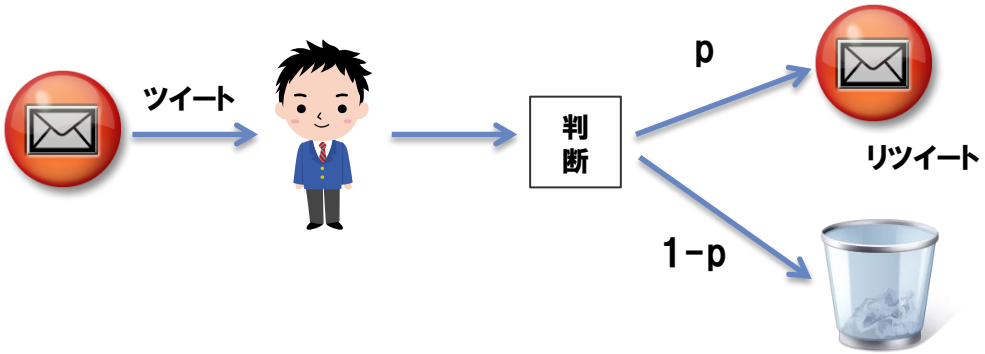
単純なルール



複雑な結果

ツイッターの伝播シミュレーション

- ツイッターユーザーの行動パターンにも様々なタイプがある。（フォロー数、フォロワー数、ツイート数、ツイート率、リツイート率、リツイートの遅延分布など）



- このような情報をもとに、ある特定のメディア（それを視聴しているユーザ層）に情報を提供することで、その内容に関する事をツイートするが、そのメッセージがどのようにネットワーク上に伝播するかを、シミュレートする事ができる。
- 自分の知っている人からのメッセージであれば、注目するし、信用もする。このようなSNS上における、新しい情報伝播モデルを分析できれば、例えば、企業の広告キャンペーンの立案などにも役立つ。

ユーザーコンファレンス2013

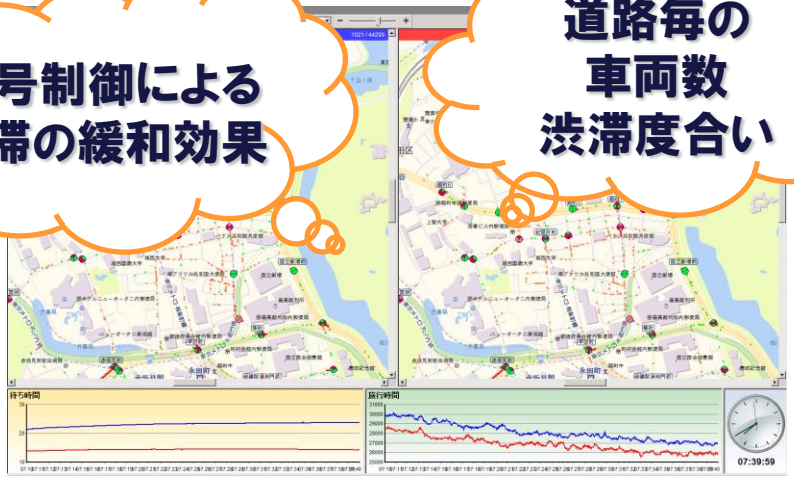
エージェントシミュレーションの事例

交通シミュレーション

- 大規模な交通シミュレーションで、都市全体に影響する交通計画やオリンピック等のイベント時の影響を予測、緩和策を検討

信号制御による渋滞の緩和効果

道路毎の車両数渋滞度合い



100万台規模の大規模処理

- HPCベースからIAベースの分散処理

低コストの渋滞予測

- 大量GPSを束ねることによる将来予測

柔軟な信号制御

- 混雑状態やセンサ設置位置のバリエーションに対応した動的信号制御

シミュレーション シナリオ

道路・交差点 接続情報

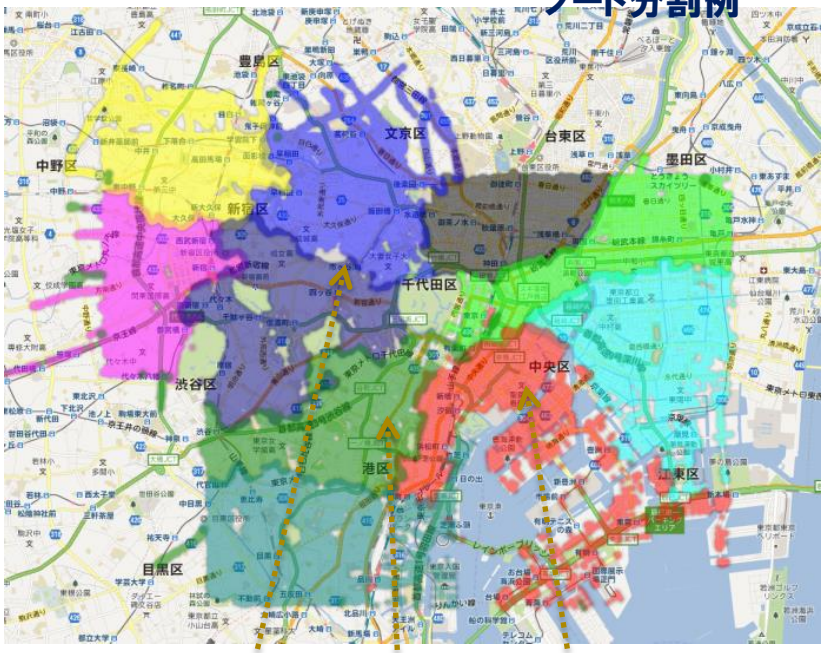
GPS プローブ

大量のGPSプローブを活用した渋滞予測・制御

交通シミュレーション

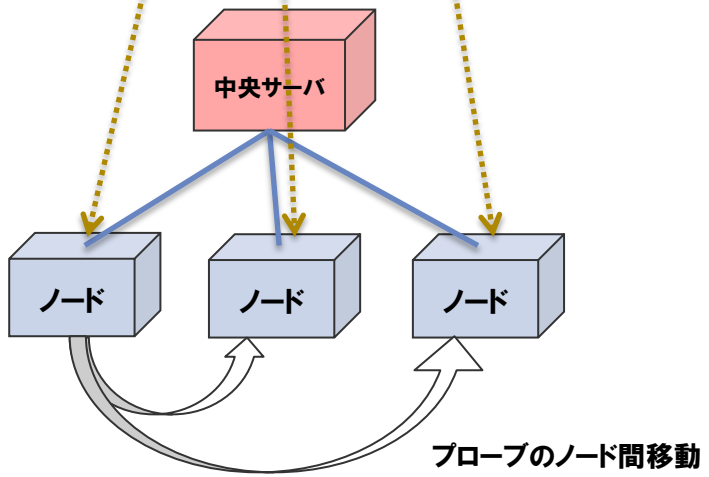
100万台規模のシミュレーションを行うために、道路ネットワークをメッシュ分割し、分散シミュレーションを行う。
分割される道路の交通量を最小化し、且つ、分割後のノードに含まれる車両台数が均等になるように分割

ノード分割例



分散処理部分はPythonを用いて実装
S⁴とのシームレスな連携が可能

個々のノードでは
S⁴を用いたシミュレーションを実行



まとめ

- S^4 Simulation System
  - 特徴
    - GUIによる簡単なモデリング
    - psim言語による柔軟なモデリング
    - 離散、連続の Hybrid なシミュレーションにも対応
  - 新機能
    - より柔軟で拡張性の高いエージェントシミュレーションのモデリングが可能になった。エージェントのモデリング方法には大きくわけて2タイプある。
      - 同期エージェント
      - 非同期エージェント
    - エージェントシミュレーションのモデリングにおいて必要となる、エージェントが属す環境、たとえばユークリッド空間、グラフ空間などはあらかじめ用意されており、ユーザはエージェントロジックのモデリングに専念することができる。
    - また、環境ごとに基本的なビューも用意されており、容易に可視化が行える。必要に応じ、シミュレーションモデル独自のビューを実装することも可能である。