生成AIを活用した製造・物流領域における デジタルツイン自動構築

MSIISMコンファレンス 2026　2026年10月16日
株式会社NTTデータ数理システム
シミュレーション＆マイニング部　豊岡 祥

© 2026 NTT DATA Mathematical Systems Inc.

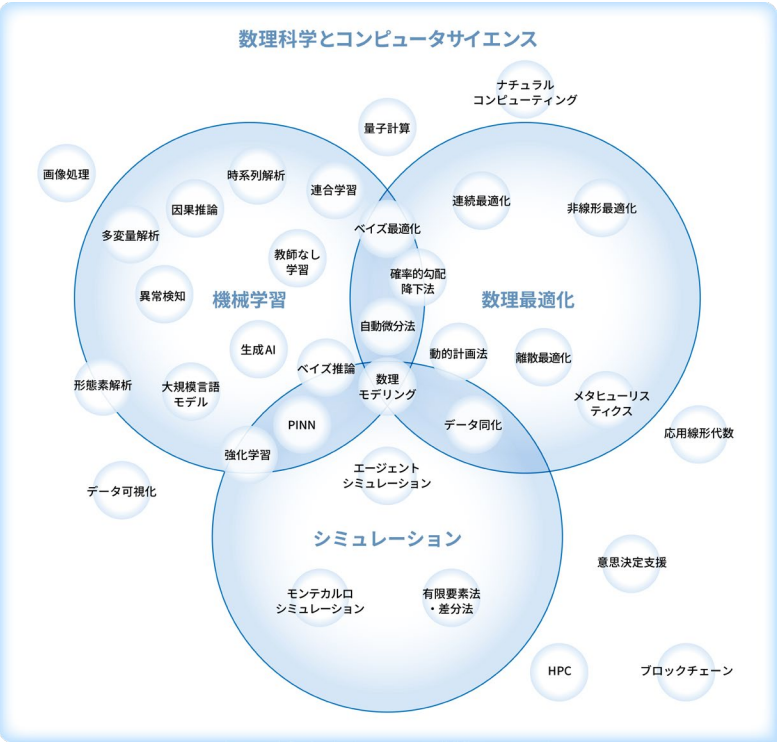
自己紹介



豊岡祥　株式会社NTTデータ数理システム
シミュレーション＆マイニング部 グループリーダー

「シミュレーション・数理最適化・機械学習」を活用した受託開発

広告配信の効果予測と最適化、配送経路最適化（ラストワンマイル）
ECサイトのレコメンデーションモデル構築、法人営業データ分析
製造ラインシミュレーション、交通シミュレーション、感染症シミュレーション



S⁴ Simulation System の開発

2D人流シミュレーションに適した移動経路計算アルゴリズム
可視化ダッシュボード、シミュレータと強化学習との接合

執筆

Python 講座 https://www.msiism.jp/column/python-code-reading/
XAI（説明可能なAI）――そのとき人工知能はどう考えたのか？
Python入門 AtCoder Programming Guide for beginners

© 2026 NTT DATA Mathematical Systems Inc.

生成AIを活用した製造・物流領域におけるデジタルツイン自動構築

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

    1. AIによるシミュレータ開発

    2. AI開発のその先

        AIによるシミュレーションの活用・運用

        AIを内包するシミュレーション

3. まとめ

シミュレーション→ 他の技術要素（機械学習・数理最適化等）
に置き換えて聞いていただいても問題ありません

© 2026 NTT DATA Mathematical Systems Inc.

生成AIを活用した製造・物流領域におけるデジタルツイン自動構築

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

    1. AIによるシミュレータ開発

    2. AI開発のその先

        AIによるシミュレーションの活用・運用

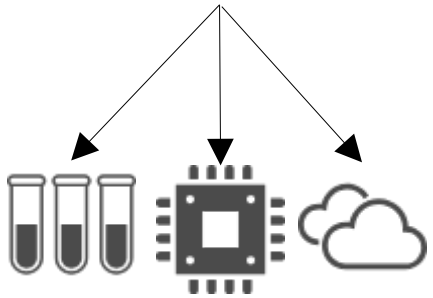
        AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

本講演で念頭に置くシミュレーション

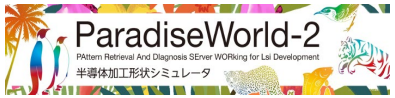
シミュレーション ＝ **計算機上で現実を再現する技術.**

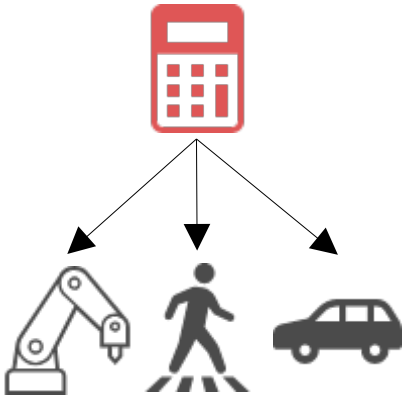


特化型シミュレーション:

扱いたい対象ごとに特化したシミュレータを構築する

- 化学プラントの制御シミュレーション
- 半導体製造加工シミュレーション
- 気象予測シミュレーション…





汎用シミュレーション:

シミュレーションの汎用部品を準備しておき、組み合わせる

- 離散イベントシミュレーション（製造・物流）
- エージェントシミュレーション（人流・交通）

S4 Simulation System

次ページから例を紹介します

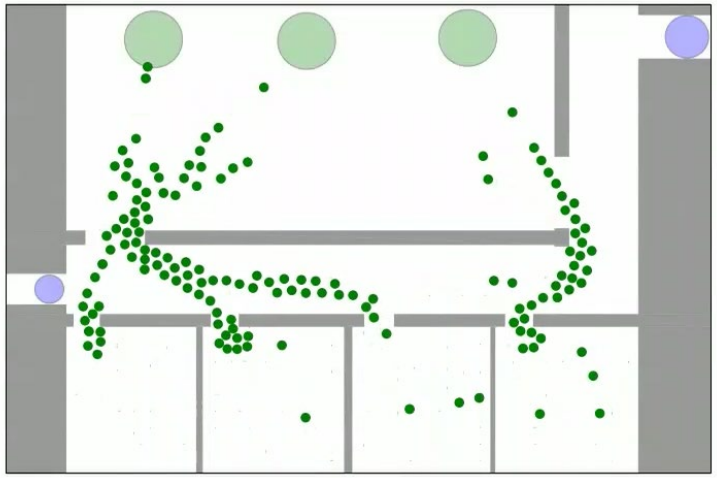
© 2026 NTT DATA Mathematical Systems Inc.

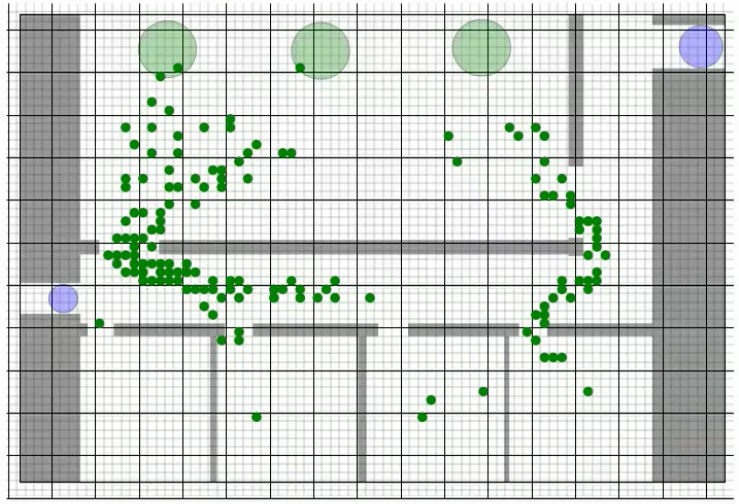
NTT DATA

人流・交通シミュレーション

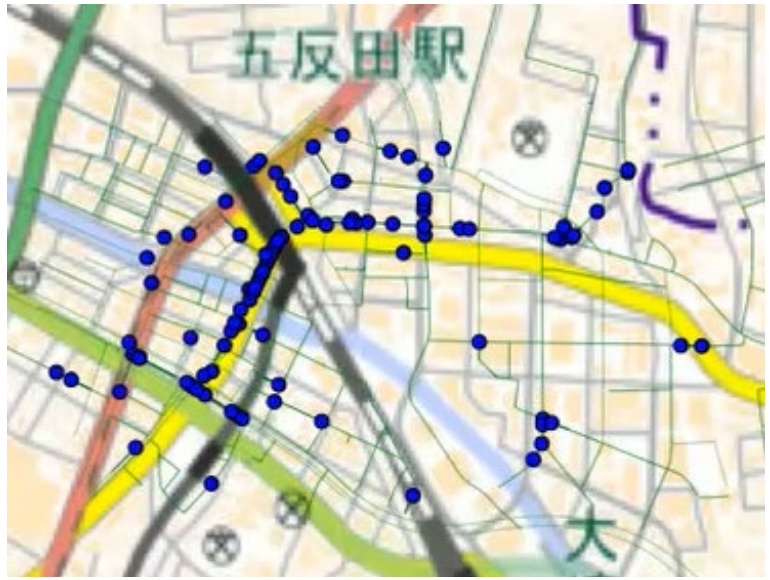
人や車の流れを再現する技術
エージェントシミュレーション ＝ エージェントの行動ルールを記述

人流シミュレーション





交通シミュレーション



出典: 国土地理院ウェブサイト

© 2026 NTT DATA Mathematical Systems Inc.

人流・交通シミュレーション

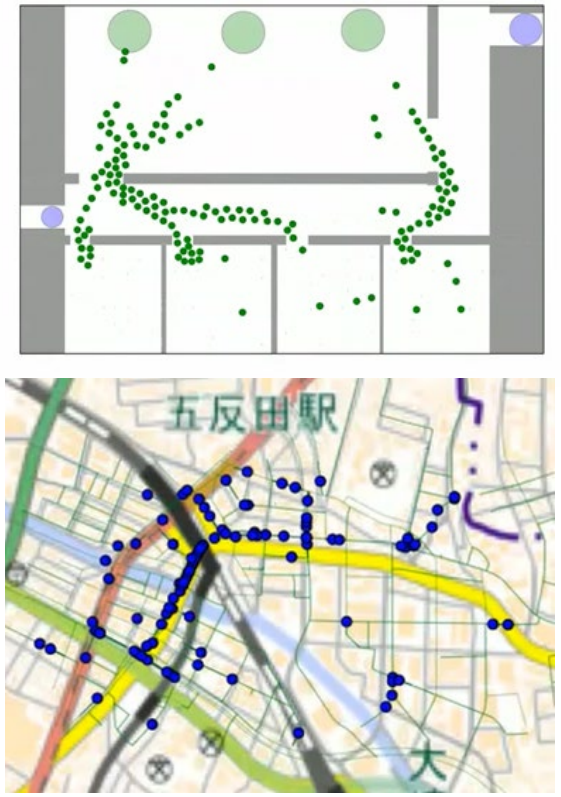
人や車の流れを再現する技術

入力

地図データ
- 移動空間/道路NW
- 障害物/通行止め

ODデータ
- 出発時刻
- 出発地
- 到着地
- 人数(台数)

シミュレーション



出力

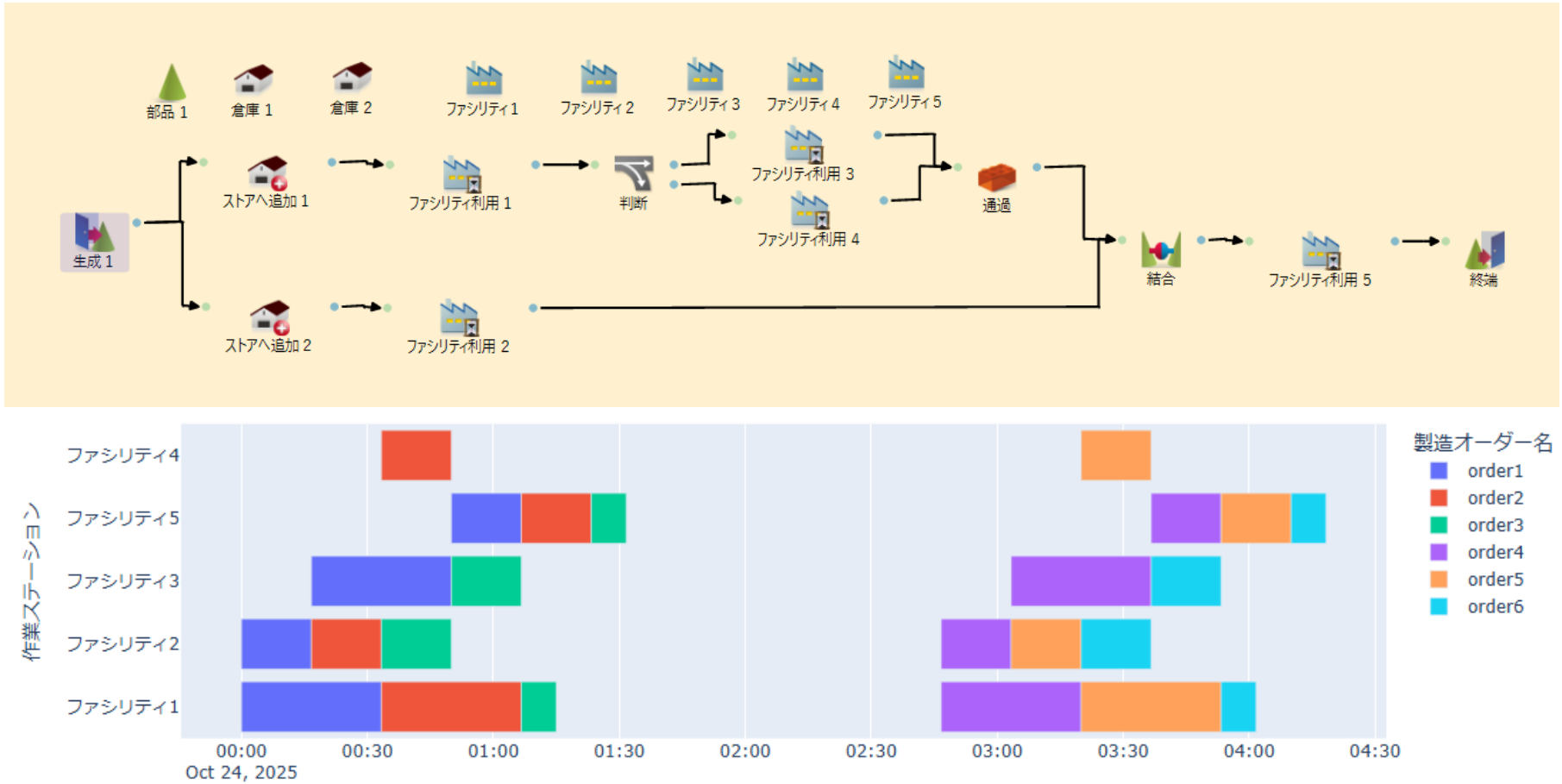
混雑データ
- 混雑箇所
- 混雑時間帯
- **渋滞長・渋滞量**

移動データ
- **所要時間**

© 2026 NTT DATA Mathematical Systems Inc.

製造ラインシミュレーション

工場の製造ラインにおける**機械の動作や部品の流れ**を再現する技術
離散イベントシミュレーション ＝ イベントの発生条件を記述



© 2026 NTT DATA Mathematical Systems Inc.

製造ラインシミュレーション

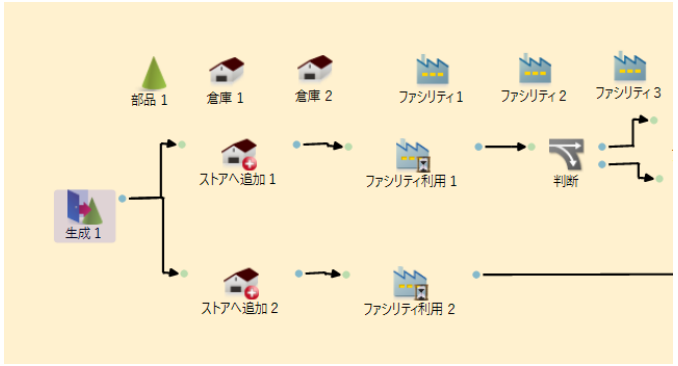
工場の製造ラインにおける**機械の動作や部品の流れ**を再現する技術

入力

設備データ
- 作業機械
- シフト
- 処理時間

生産計画
- 投入順序
- 分岐・合流条件 …

シミュレーション



出力

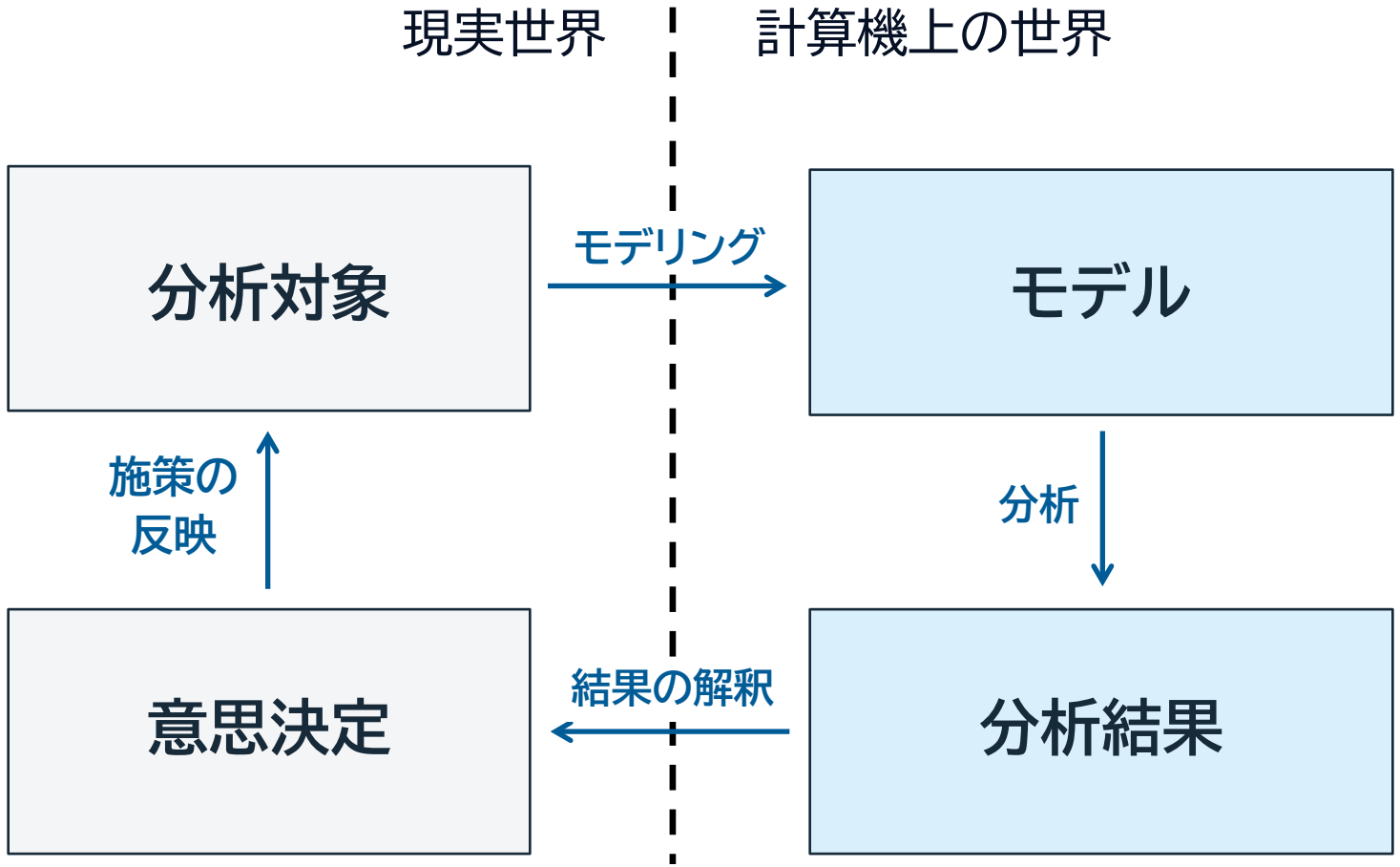
製造結果
- **生産効率**
- ガントチャート

分析
- 機械の**稼働率**
- 滞留数・待ち時間
- **ボトルネック**可視化

© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインとは？

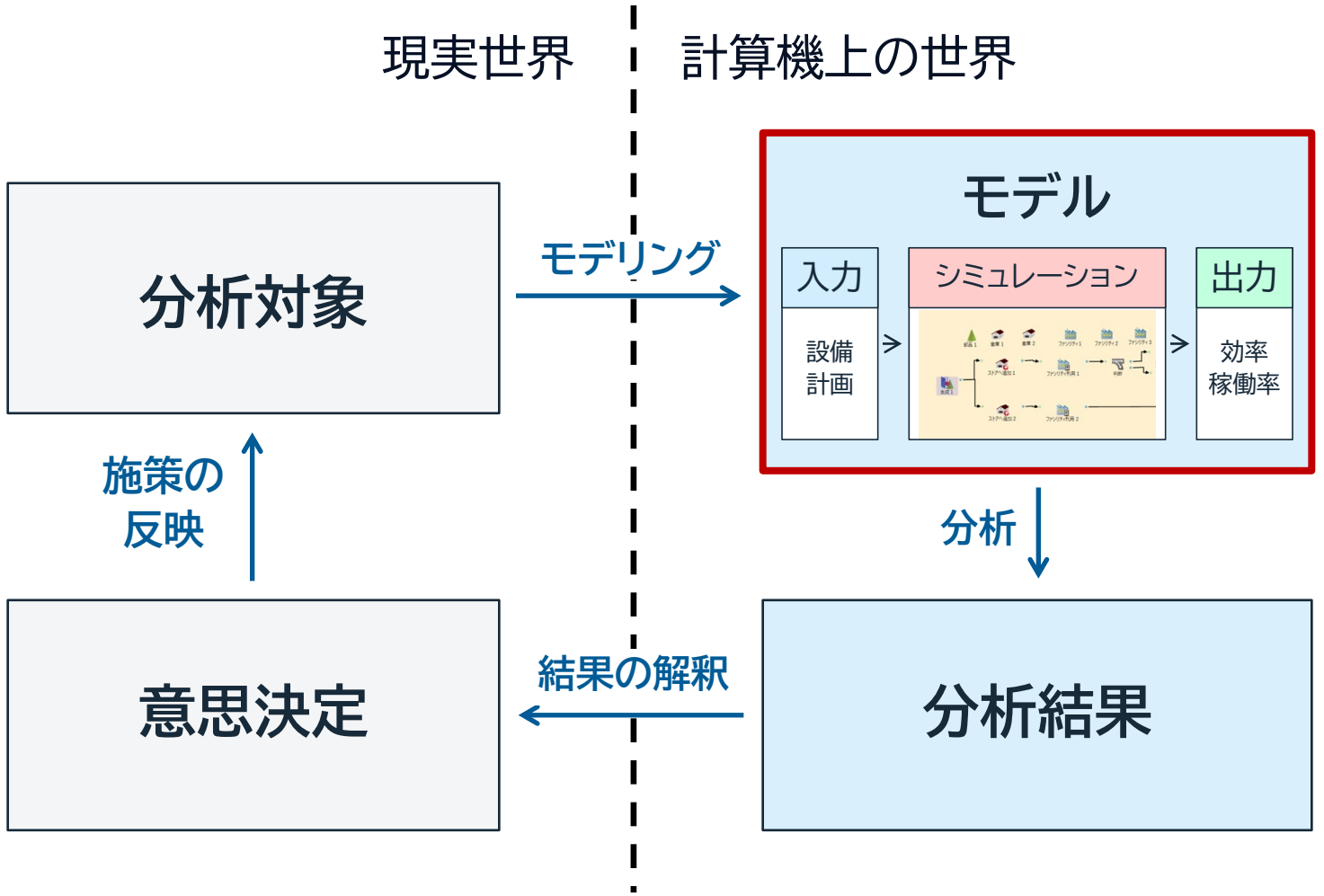
分析対象を計算機上のモデルで表現(**モデリング**)
→ 計算機上の**分析**（**アルゴリズム**）→ 結果を現実の意思決定に反映



© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるシミュレーションの活用

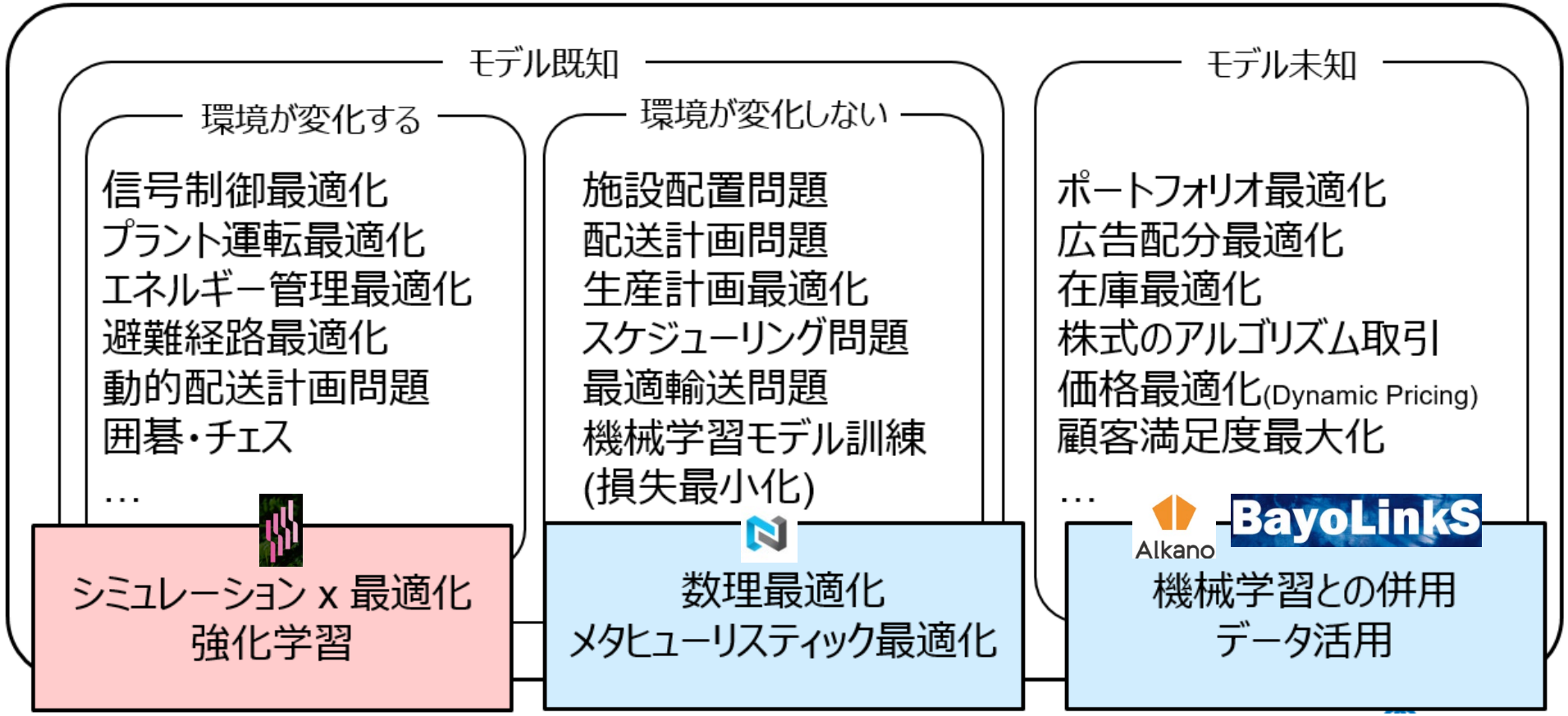
シミュレーションは主要な**モデリング**技術のひとつ



© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるシミュレーションの活用

（脱線）シミュレーション以外のモデリング技術：数理最適化・機械学習など
使い分けが重要（昨年度講演↓）



© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるシミュレーションの活用

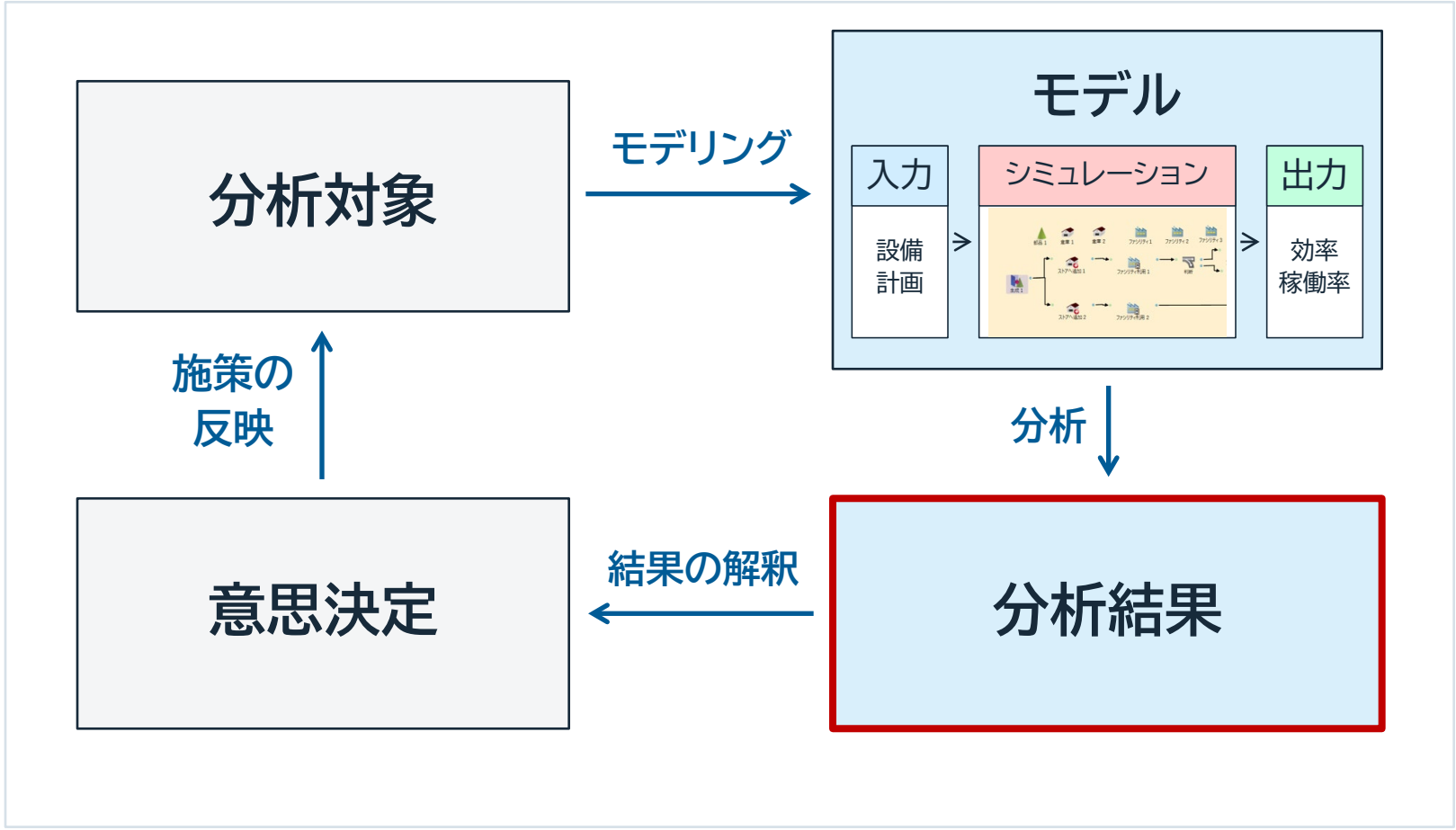
シミュレーションモデルを使った**分析**のパターン

パターン① what-if分析

条件を変えて実行し、将来予測・リスク評価に活用

パターン② 最適化

結果を最も良くする入力条件を探索する。



© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるシミュレーションの活用

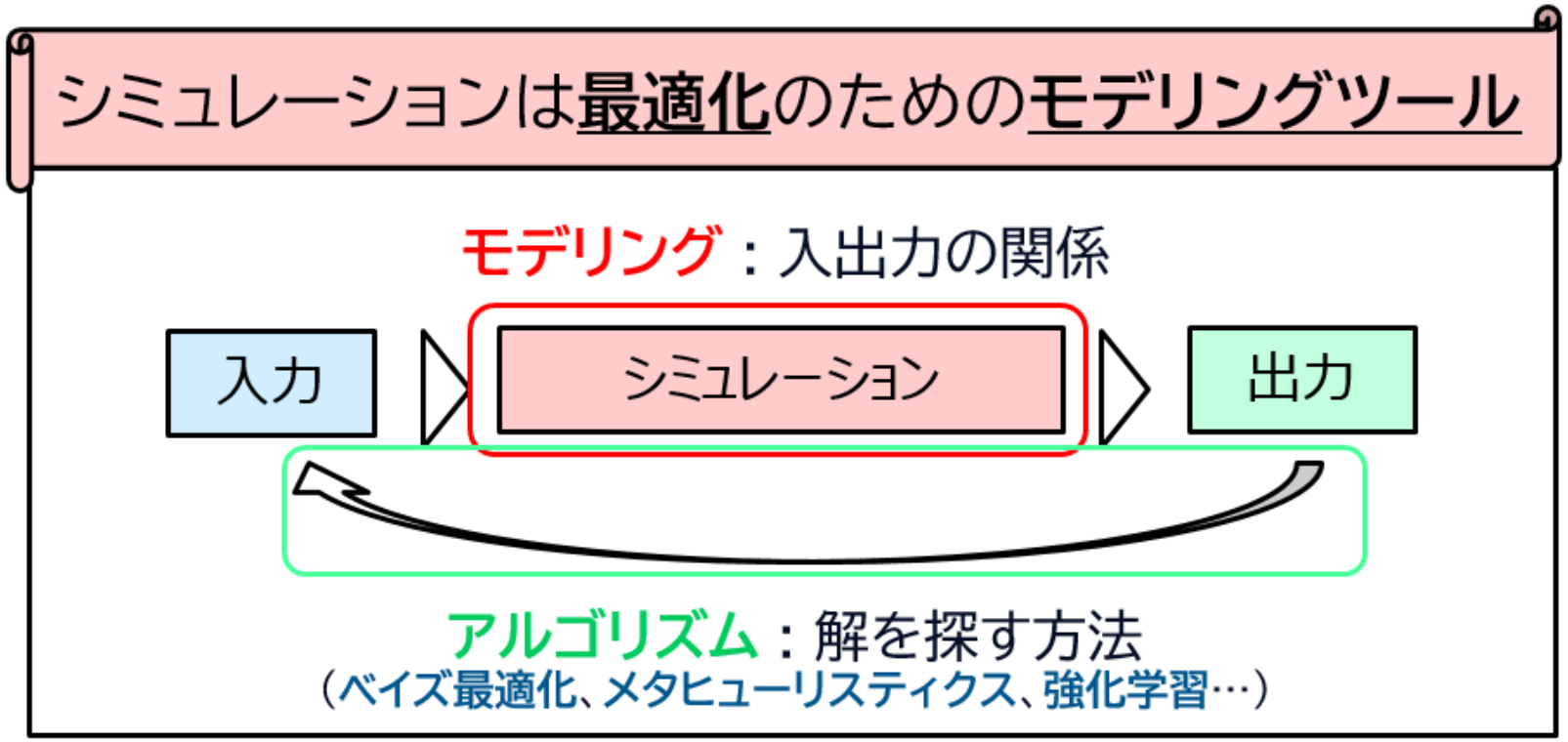
（脱線2）Q. シミュレーションで最適化？最適化は数理最適化じゃないの？
A. モデリングとアルゴリズムの組み合わせ次第。色々な最適化の形があります
（昨年度講演）

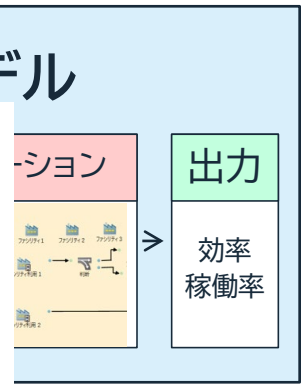
パターン① wha

条件を変えて実行
測・リスク評価に

パターン② 最適

結果を最も良くて
を探索する。





結果

© 2026 NTT DATA Mathematical Systems Inc.

活用例：構内物流デジタルツイン

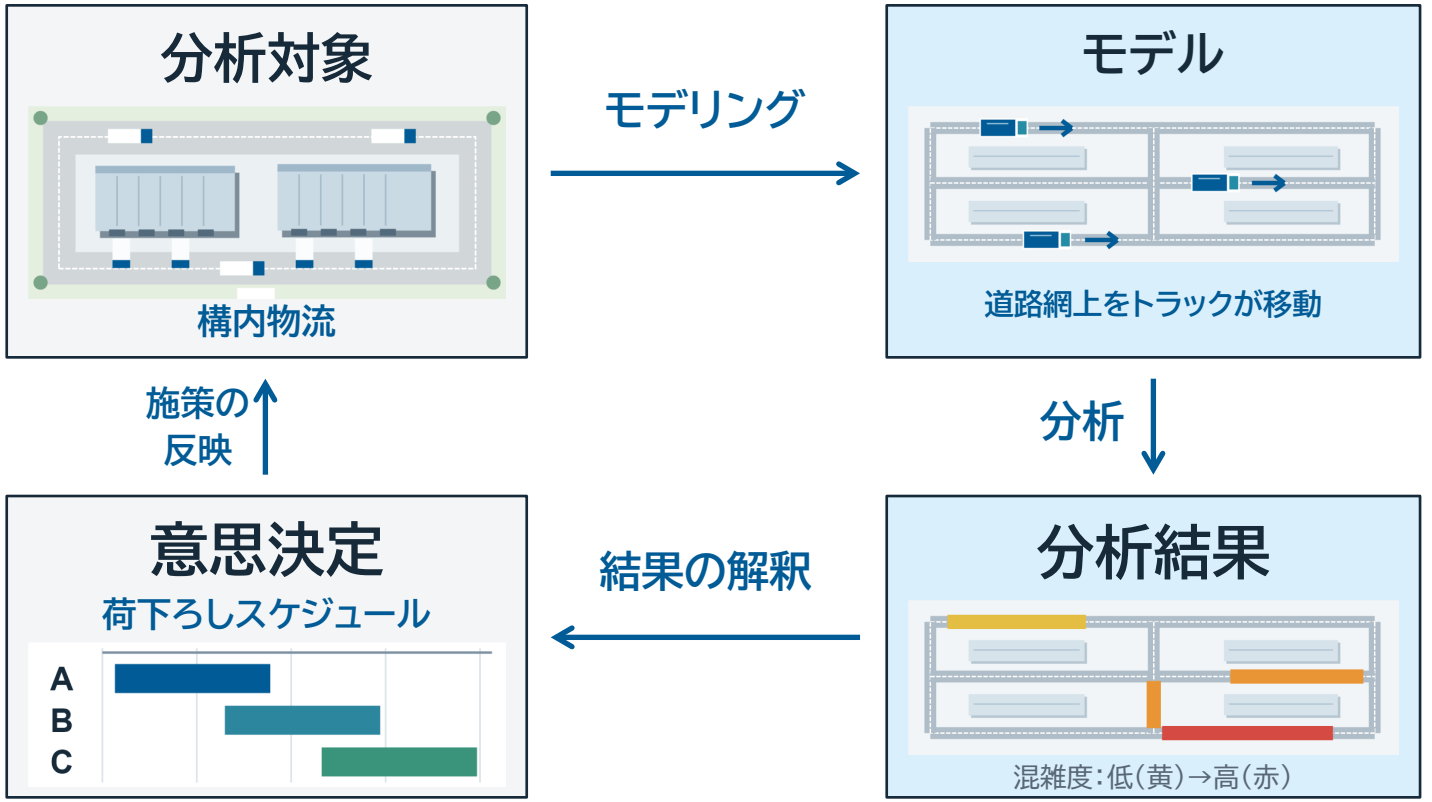
部品運搬トラックの移動・荷下ろしの再現～評価～最適化

① データ収集
到着予定・荷下ろし時間 / 納期 / 道路網

② モデリング
トラックの移動・荷下ろし・製造ライン投入等
を再現するシミュレータ

③ What-if分析 / 最適化
一部トラックが遅延した場合の**納期影響予測**
荷下ろしスケジュール改善による**混雑最小化**

④ 意思決定
施策(スケジュール変更)へ反映



© 2026 NTT DATA Mathematical Systems Inc.

活用例：構内物流デジタルツイン

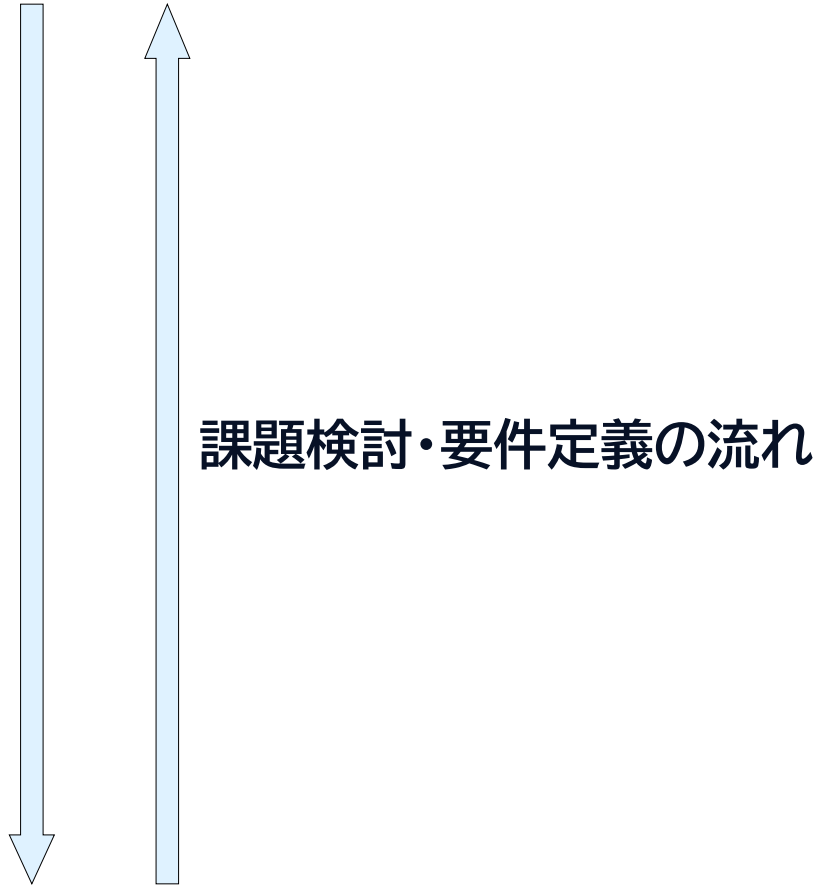
（脱線3）検討フェーズと検証フェーズは流れが逆

① データ収集
到着予定・荷下ろし時間 / 納期 / 道路網

② モデリング
トラックの移動・荷下ろし・製造ライン投入等
を再現するシミュレータ

③ What-if分析 / 最適化
一部トラックが遅延した場合の**納期影響予測**
荷下ろしスケジュール改善による**混雑最小化**

④ 意思決定
施策(スケジュール変更)へ反映



© 2026 NTT DATA Mathematical Systems Inc.

本日の内容

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

   1. AIによるシミュレータ開発

   2. AI開発のその先

      AIによるシミュレーションの活用・運用

      AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

シミュレーション活用の課題

① モデルの構築・更新の負担

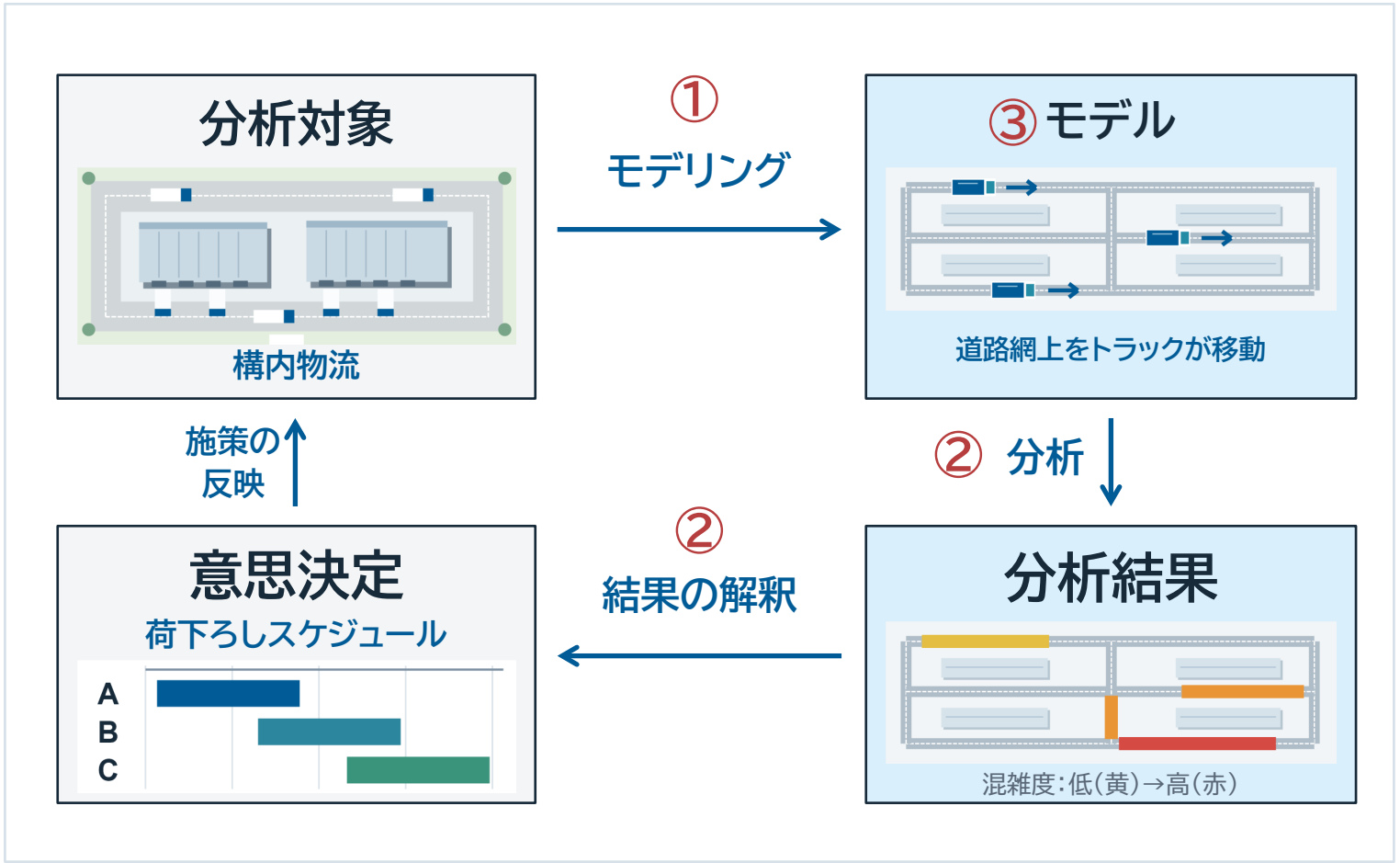
データ収集・現場ロジックの記述
（大量のif-thenルール）

② 実験・解釈の負担

実験→解釈→修正のサイクルを
多く回す必要

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
要素が存在



© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるAI活用

① モデルの構築・更新の負担

データ収集・現場ロジックの再現
（大量のif-thenルール）

→ **AIによるシミュレータ開発**
（AI develops simulator）

② 実験・解釈の負担

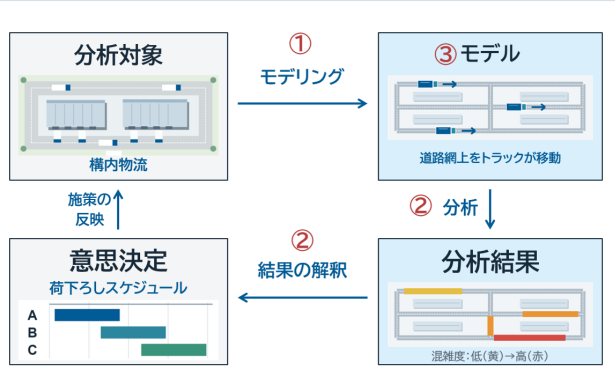
実験→解釈→修正のサイクルを
多く回す必要

→ **AIによるシミュレーション活用・運用**
（AI operates simulation）

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
要素が存在

→ **AIを内包するシミュレーション**
（simulator includes AI）



© 2026 NTT DATA Mathematical Systems Inc.

本日の内容

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

   1. AIによるシミュレータ開発

   2. AI開発のその先

      AIによるシミュレーションの活用・運用

      AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるAI活用

① AIコーディングは強い
② ツール整備が重要

① モデルの構築・更新の負担

データ収集・現場ロジックの再現
（大量のif-thenルール）

→ **AIによるシミュレータ開発**
（AI develops simulator）

② 実験・解釈の負担

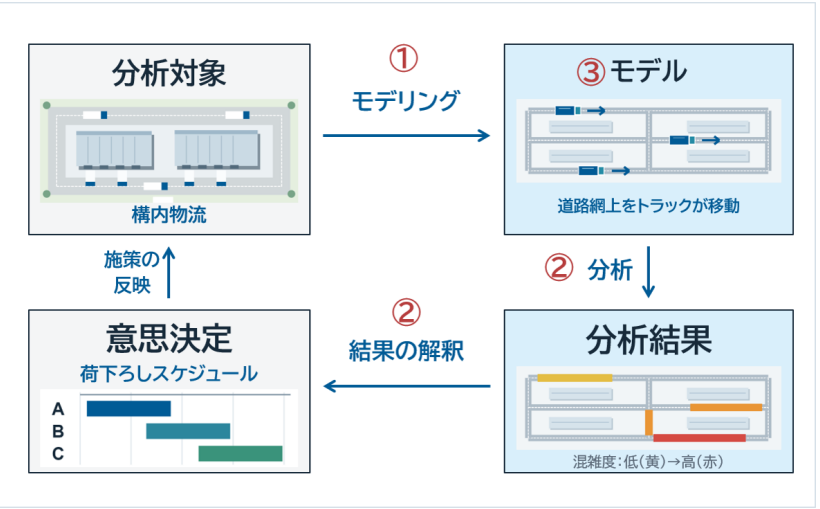
有益な示唆を得るには多くの
試行錯誤が必要

→ **AIによるシミュレーション活用・運用**
（AI operates simulation）

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
現実世界の要素が存在

→ **AIを内包するシミュレーション**
（simulator includes AI）



© 2026 NTT DATA Mathematical Systems Inc.

ソフトウエア開発の流れ

要件定義 → **設計** → **実装** → **テスト**

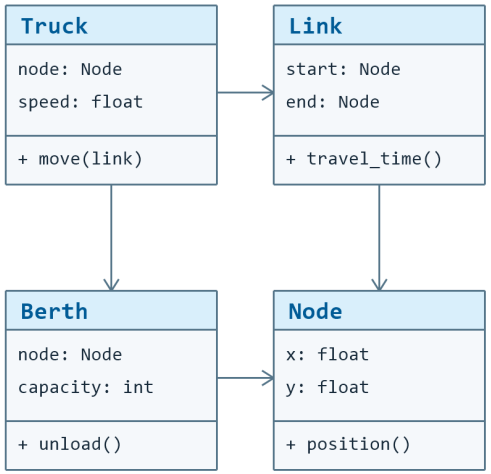
シミュレーションの
構成要素と動作仕様
を決める。



- トラック
- ノード・リンク
- バース
  etc…

プログラムの構造を
決める。

Class diagram



設計に沿って
プログラムを作る。

Python

```python
class Truck:
    def __init__(self, node):
        self.node = node

    def move(self, link):
        self.node = link.end


# Move along a link
truck.move(link)
```

要件どおりに
動くかを確認する。

移動テスト(模式例)

入力:AからBへ移動

A → B

期待値	実行結果
到着先:B	到着先:B

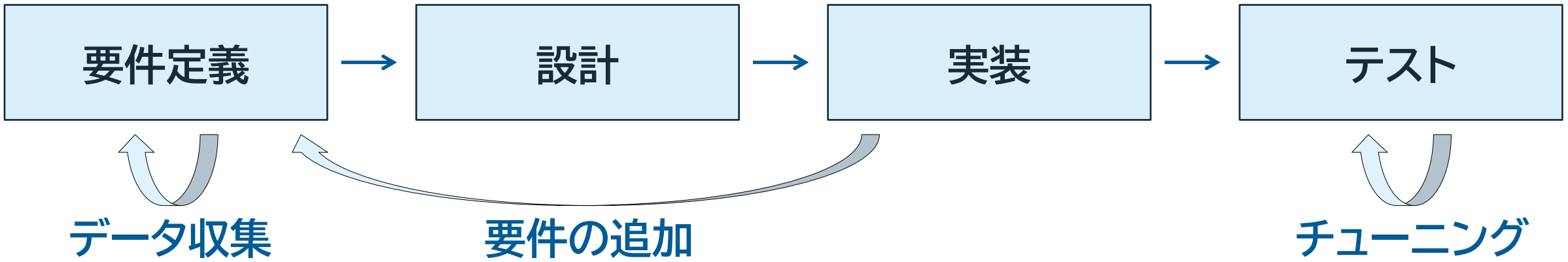
比較して一致を確認

荷下ろし・受入制約も確認

© 2026 NTT DATA Mathematical Systems Inc.

シミュレータ開発の流れ

シミュレータ開発は特に**データ収集 / 要件の追加 / 精度向上のためのチューニング**に時間がかかることが多い。
→ モデルの構築・更新の負担軽減のためには**プロトタイプ構築の高速化**が不可欠

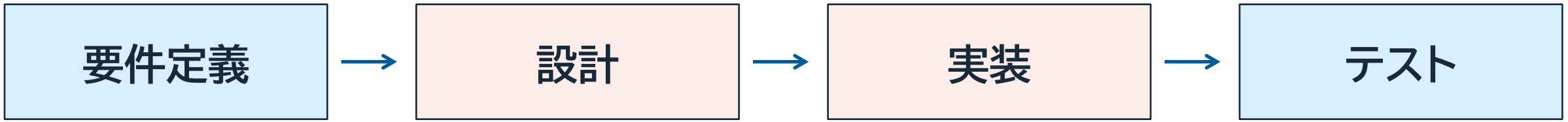


大規模なシミュレータ開発には
数カ月～年単位かかることも

© 2026 NTT DATA Mathematical Systems Inc.

AIコーディングの実力

AIは「設計→実装」に特に強い。



シミュレーションの
構成要素と動作仕様
を決める。

プログラムの構造を
決める。
論理的思考力

設計に沿って
プログラムを作る。
実装能力

要件どおりに
動くかを確認する。

© 2026 NTT DATA Mathematical Systems Inc.

ノーコード／ローコードからAIコードへ

アプリケーション開発(シミュレーションに限らない)の効率化の変遷

ハイコード

Pythonエディタ

simulation.py

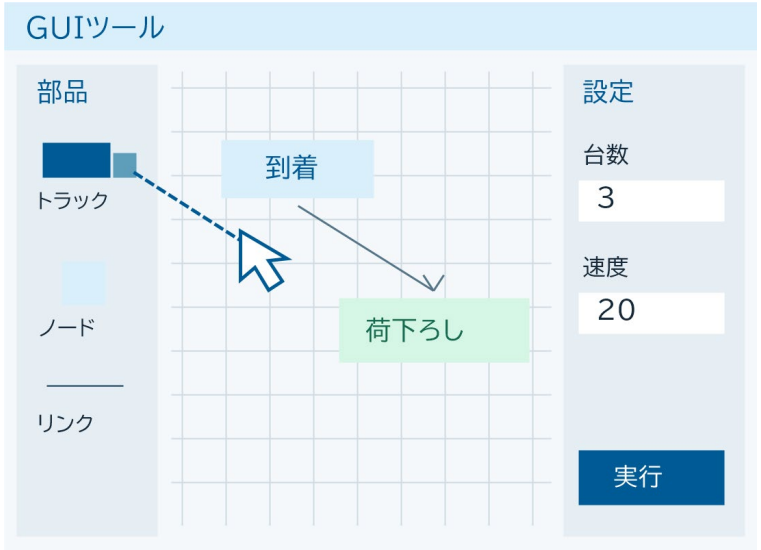
```
1  def simulate(events):
2      time = 0
3      while events:
4          event = events.pop(0)
5          time = event.at
6          event.run()
7      return time
```

コードを記述･修正

人がプログラムを記述

複雑なものを構築可能

ノーコード／ローコード



ドラッグして配置･設定

スキルが無くても構築可能
ハイコードとの併用も

AIコード

AIエージェント

人の指示

荷下ろし待ちを含む
物流モデルを作って

AIがコードを生成

```
def build_model():
    model = Model()
    ...
```

指示 → 生成 → 確認･修正

AIに指示して生成

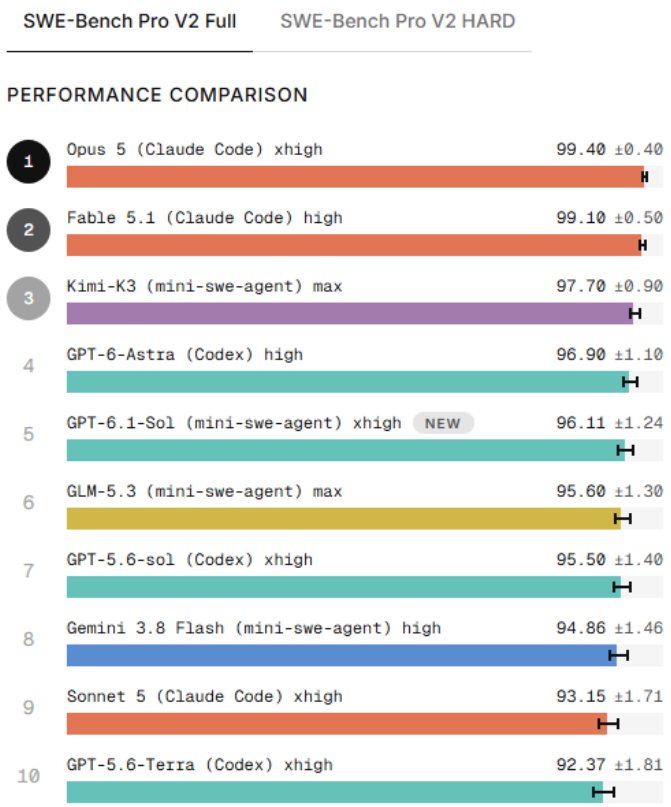
スキルが無くても複雑なものを構築可能…？

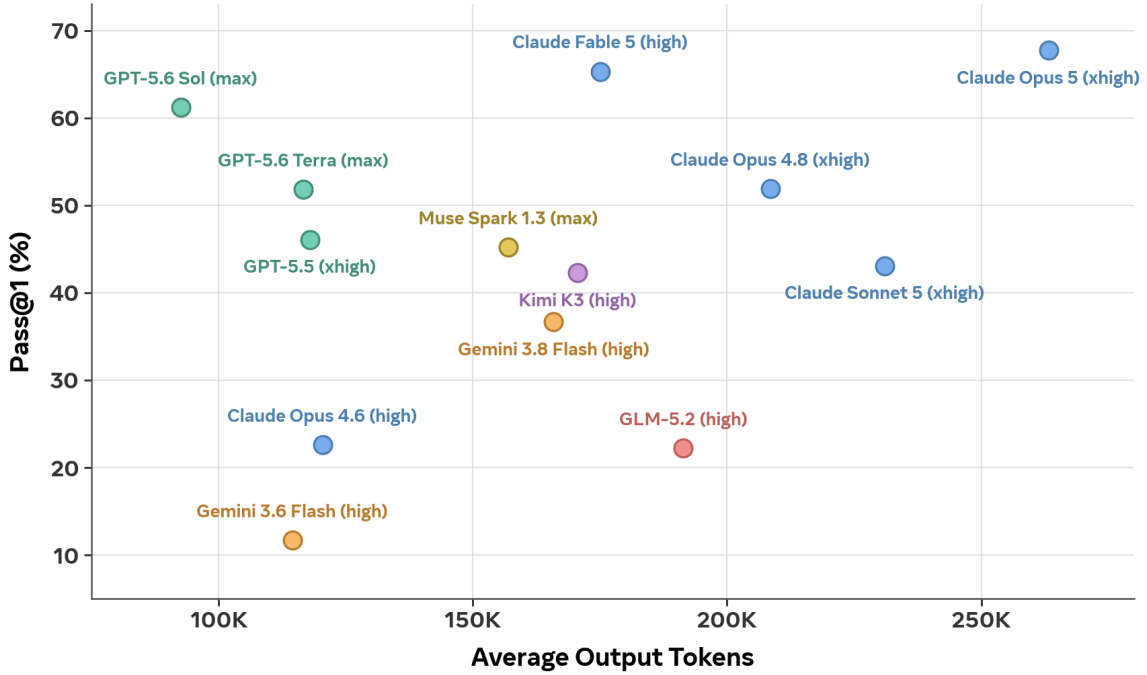
© 2026 NTT DATA Mathematical Systems Inc.

AIコーディングの実力：実装能力

SWE-Bench Pro V2, E2E-SWE 等のベンチマークで高い成績

- 実プログラムの改修や機能追加を行うテストにおいて 95% 以上成功
- 仕様をもとに一からプロジェクト構築するテストにおいて60%以上成功





© 2026 NTT DATA Mathematical Systems Inc.

AIコーディングの実力：論理的思考力

登場から2-3年のあいだに世界トップレベルの競技プログラミングコンテストで、人間を上回る水準に到達

- ～2024年: 生成AIの台頭. ビギナー向けコンテストから順次、**生成AI利用が禁止**
  - 最初の対応から1年ほどで全コンテストでの利用が禁止
- 2025年: 10時間の人類最高峰コンテストで人類1位, AI2位
- 2026年: 同じコンテストの2部門でAIが圧倒的に1位

1	Psyho	1,812,272,558,909 594:53
2	OpenAI	1,654,675,725,406 582:56



←2025　2026→

順位	ユーザ	得点	A	B	C	D	E
1	OpenAI	8300 (7) 390:36	900 30:05	900 32:11	1500 28:48	2500 (2) 174:24	2500 (5) 355:36
2	tour1st	4300 (3) 412:25	900 (1) 397:25	900 58:13	-	2500 (2) 362:29	-
3	jiangly	1800 111:07	900 27:21	900 111:07	-	-	-
4	turmax	1800 (1) 147:57	900 (1) 102:16	900 142:57	-	-	-
5	gblaasnicse	1800 160:48	900 160:48	900 42:16	-	-	-

1	OpenAI	1,999,878,828,865 2013:54	50,000,000,000 (61) 2013:54	1,999,878,828,865 2013:54
2	Shun_PI	481,395,927,450 1960:36	12,511,595,075 (36) 1960:36	481,395,927,450 1960:36
3	yosupo	403,384,272,283 2038:31	8,411,185,061 (23) 2038:31	403,384,272,283 2038:31
4	eijirou	324,056,684,900 2033:40	6,912,698,281 (21) 2033:40	324,056,684,900 2033:40
5	MathGorilla	308,336,644,874 2011:11	6,938,636,630 (30) 2011:11	308,336,644,874 2011:11

© 2026 NTT DATA Mathematical Systems Inc.

AIコーディングの実力：論理的思考力

その他、多くのベンチマークや数学研究における成果

- Humanity’s Last Exam（Diamond）
  - あらゆるジャンルの難問を集めたデータセット（常人はほぼ解けない）
  - 最近のモデルは50%以上のスコア
- GPQA Diamond
  - 生物学・物理学・化学の大学院レベルの難問. 人間の専門家で65%程度のところを、95%以上のスコア
- 国際数学オリンピックにおいて金メダル相当(2025)→ 満点相当(2026)の実力
- AlphaEvolve による行列積演算アルゴリズムの改善（2025）
- OpenAI が700を超える数学の論文を発表（2026/10）
  - 分野における重要な未解決問題を含む
  - 内容の妥当性については検証を待つ部分も多い

© 2026 NTT DATA Mathematical Systems Inc.

AIコーディングの課題

動作するプログラムを作成することは容易。
人間による妥当性の担保が課題。

① ソースコードは要件どおりか

要件(例)

バースが
満車なら待つ

☐ 移動
☐ 荷下ろし
☐ 受入制約

照合

truck.py / route.py / berth.py

```
class Berth:
    def request(self, truck):
        if self.is_full():
            ...
        ...
    # More code across files
```

コード量が膨大なことも！

どこに実装？　抜け・誤りは？

② テスト内容は妥当か

テストの条件 → 期待値

空きあり → 荷下ろし
実行結果と一致　OK

満車時 → 待機
このケースも必要では？　？

条件と期待値が、要件に合っているか

網羅性は十分？「やったふり」はないか？

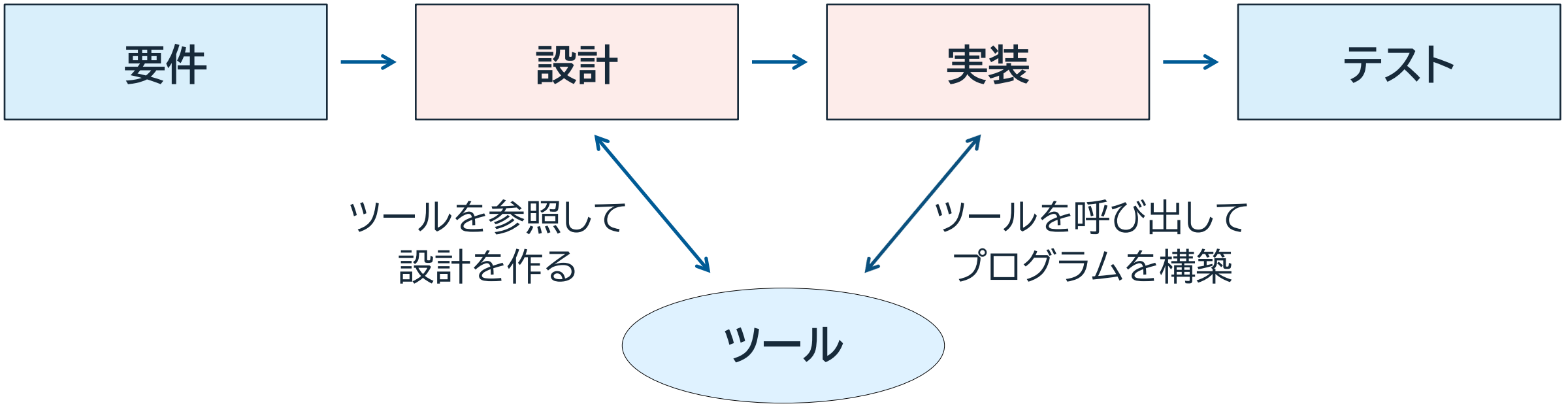
③ ステークホルダーに説明できるか

「どのようなロジックで動作している？」「利用しているOSSは安全？」「こういう条件だと何が起こる？」

© 2026 NTT DATA Mathematical Systems Inc.

解決策: ツール呼び出しによるAIの制御

AIは外部のツールを呼び出すことができる



© 2026 NTT DATA Mathematical Systems Inc.

解決策: ツール呼び出しによるAIの制御

シミュレーション構築をS4が提供する機能に(なるべく)限定する

設計 → **実装** → **ツールによる実行**

① 要件をS4の機能に分解

トラック
＋
道路網(移動)
＋
バース(荷下ろし)

S4 Simulation System

② 表形式データに記述

バース
道路網
トラック

ID	入構時刻	利用バース
T01	08:00	B01
T02	08:15	B02
T03	08:30	B01
…	…	…

S4 Simulation System

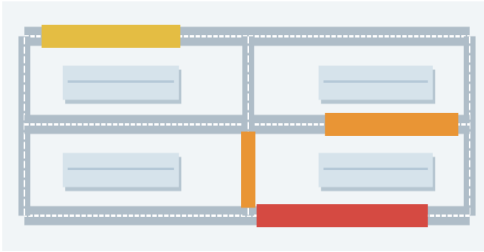
③ 現場特有のロジックだけ実装

追加コード(例)

```
def priority(truck):
    #納期が近い順に荷下ろし
    return truck.due
```

Python

④ シミュレーション実行



混雑度:低(黄)→高(赤)

S4 Simulation System

AIによる自動実行

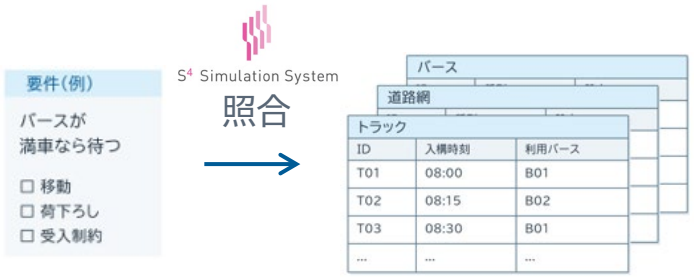
※ 表形式データによるシミュレーションの記述はS4の新機能. AI に使わせる前提で鋭意開発中！

© 2026 NTT DATA Mathematical Systems Inc.

解決策: ツール呼び出しによるAIの制御

- 安全性・妥当性確認の容易さ・説明可能性を担保
- 追加要件に迅速に対応可能（完成したものをちょっと直す、はAIにとっては容易）

① 要件との照合



Python

追加コード

例：納期順に
優先する

表形式データ＋一部の追加コードなので**確認が容易**
危険なコードが紛れ込む**リスクが小さい**

② 固有ロジックをテスト

入力条件	確認する結果
通常	納期順に選択
バース満杯	空くまで待機
不正データ	入力を検証

テストは追加コードによる固有ロジック部分に集中

③ 根拠を示してステークホルダーに説明

ロジックは？

機能・設定・追加コードを提示

OSSは安全？

依存関係・脆弱性・ライセンスを確認

この条件だと？

条件を変えて実行し、結果を提示

© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

倉庫から生産ラインに向けてAGVで部品の搬送を行うシミュレーション

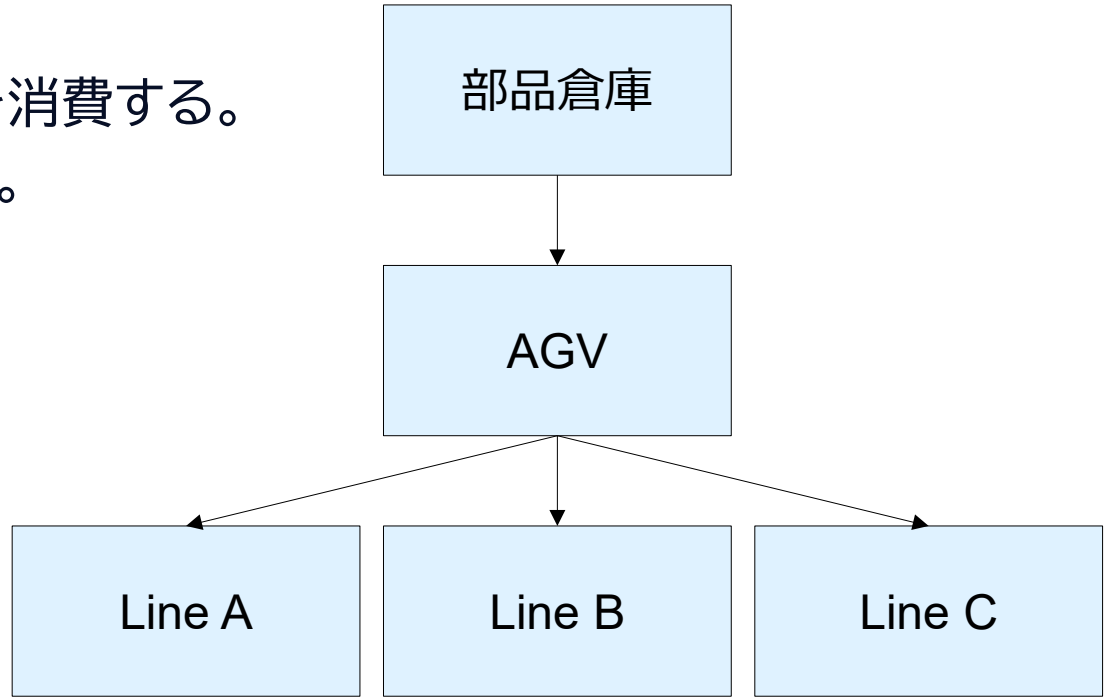
構成要素

- 部品倉庫：1か所
- 生産ライン：3ライン
- AGV：複数台

動作

- 生産ラインは一定の生産速度で稼働し、必要な部品を消費する。
- 部品在庫が補充点を下回ると搬送要求を発生させる。
- 搬送要求に対しAGV が部品を搬送

→ **必要なAGVの台数を知りたい**



© 2026 NTT DATA Mathematical Systems Inc.

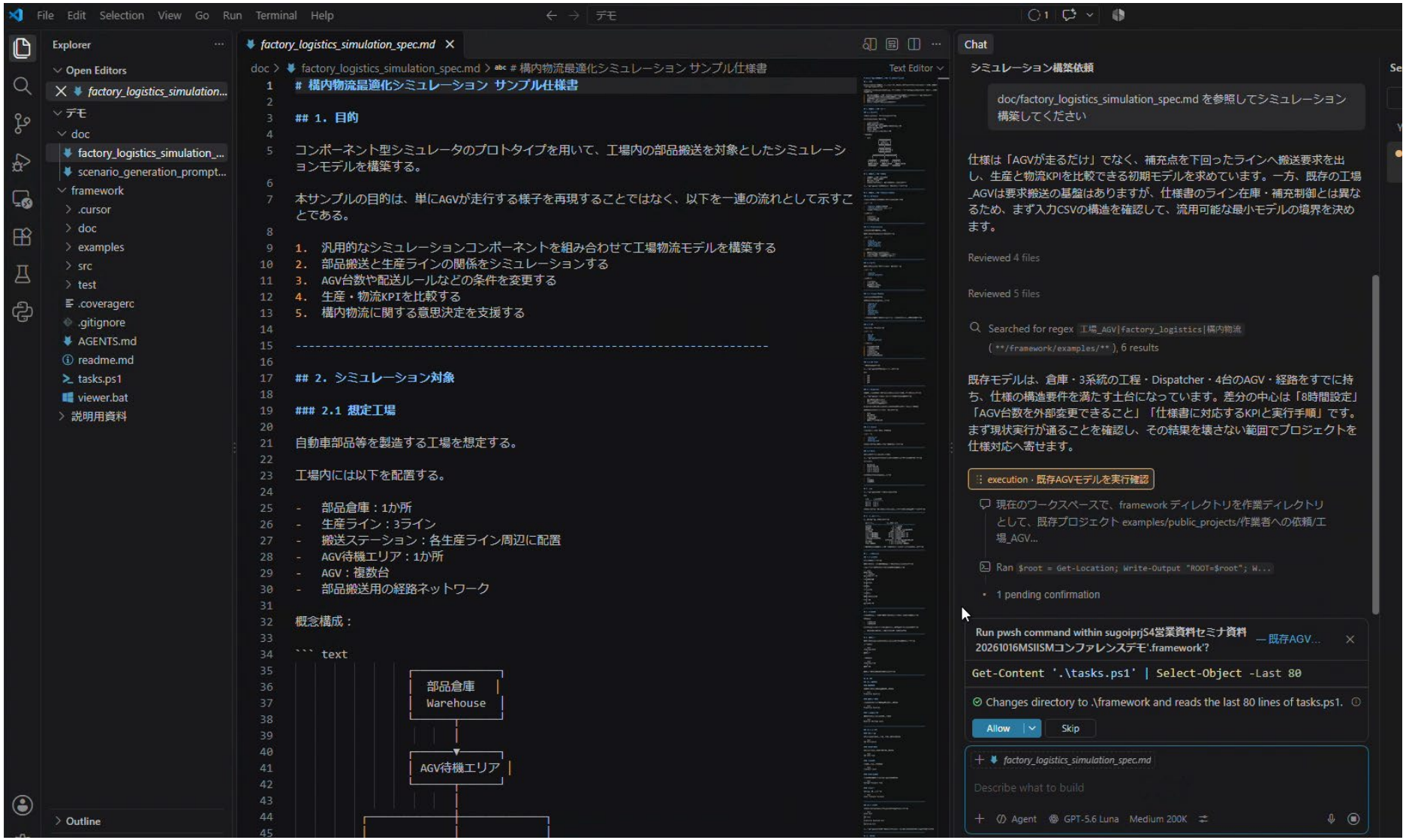
デモ: 生成AI x S4 によるシミュレーション構築



© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

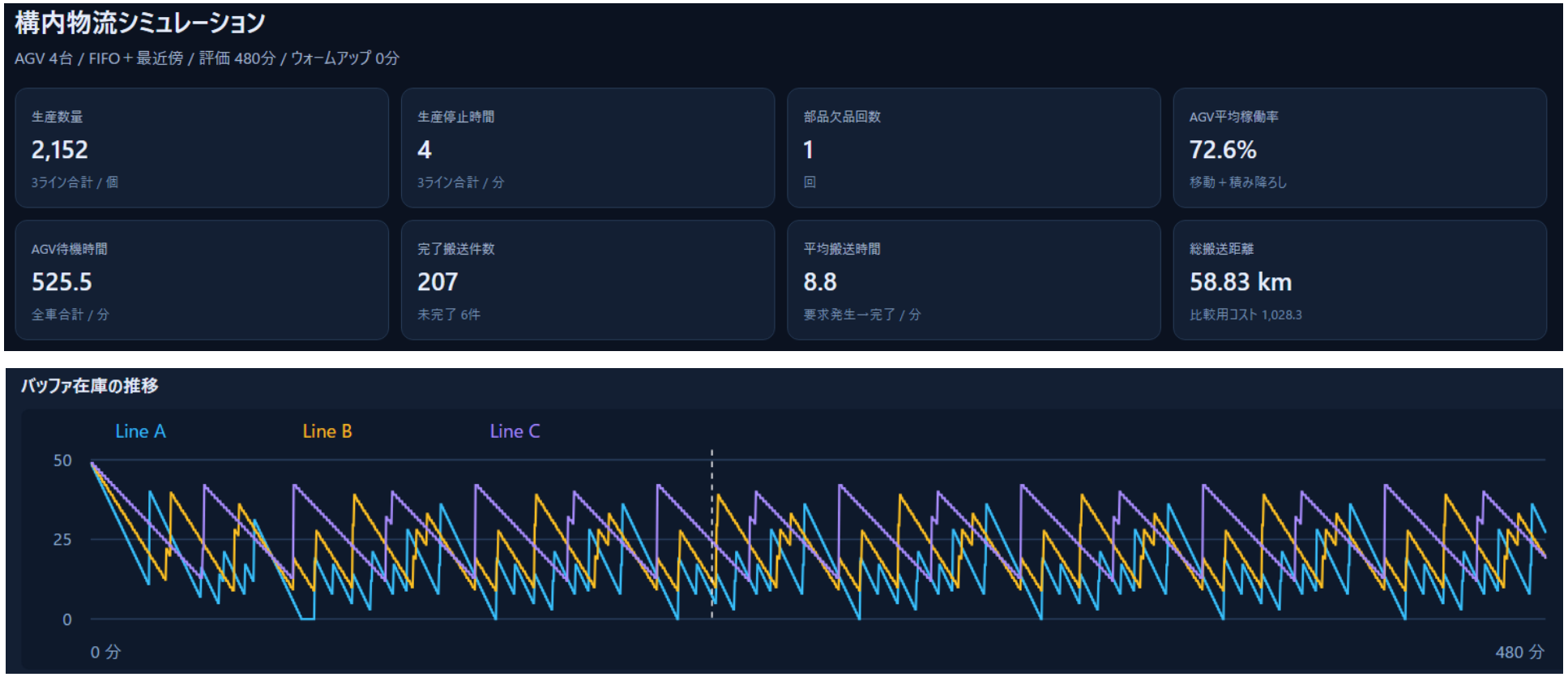
構築の様子



© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

構築結果(KPI)



© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

構築結果(アニメーション)



© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

追加実験①

AGV台数を減らして経費削減できないか。減らした場合に Line で部品が枯渇しないためにはどうすればよいか？

- AGV台数を変更する
  - 2台 / 3台 / 4台
- 配送ルールを変更する
  - 要求順 / 優先度順
  - 優先度は部品枯渇までの秒数で決定

（この実験もAIに指示してやらせることが可能）

© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

追加実験①の結果

- Line停止を無くすためには台数は4台あれば十分だが、稼働率は落ちる
- 投資対効果が高いのは3台
- 配送ルール変更の恩恵は大きくない

台数	配送ルール	製造個数	Line停止[分]	AGV稼働率[%]
2	要求順	1451	402	96.8
2	優先度順	1490	422	96.8
3	要求順	2038	63	92.7
3	優先度順	2083	41	94.2
4	要求順	2152	4	72.6
4	優先度順	2160	0	72.6

© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

追加実験②

以下のような突発事象に対して、現行の台数

- シナリオ1. 需要増加
  - シミュレーション開始後4時間経過した時点で、LineB の生産速度が1.5→2.5個/分に増加
- シナリオ2. AGV故障
  - AGV2を180～240分の間停止
- シナリオ3. 経路閉鎖
  - 300～360分、Line B直結リンクを閉鎖して迂回

（この実験もAIに指示してやらせることが可能）

© 2026 NTT DATA Mathematical Systems Inc.

デモ: 生成AI x S4 によるシミュレーション構築

追加実験②の結果

- 需要増加に伴い生産数量も最大240個増えるが、3台だと搬送が追いつかない。4台あれば需要増加の8割程度に対応できる。
- AGV故障時、3台だと3%程度、4台だと0.7%程度の生産量低下が見込まれる。
- 経路閉鎖時、3台だと2%程度の生産量低下が見込まれる。4台ではほぼ影響は無い。

シナリオ	台数	生産数量	正常時との差	合計停止時間	正常時との差
需要増加	3	2,110	+27	140.37分	+99.37分
需要増加	4	2,352	+192	20.00分	+20.00分
AGV故障	3	2,012	−71	76.17分	+35.17分
AGV故障	4	2,146	−14	7.17分	+7.17分
経路閉鎖	3	2,024	−59	76.33分	+35.33分
経路閉鎖	4	2,159	−1	0.67分	+0.67分

© 2026 NTT DATA Mathematical Systems Inc.

FAQ

Q. 適用可能性は？
A. S4が表現可能な要素（製造ライン）
待ち行列・分岐・合流等の汎用的な概念を含むため**業種を問わず適用可能**

Q. シミュレーションだけ？
A. 網羅的な条件下で実行する**一括実行**、シミュレーション結果を最良にする方法を調べる**最適化実行**、シミュレーション中の行動パターンを最適化する方策最適化（e.g. **強化学習**）等が可能

Q. AIは何が使える？
A. codex, github copilot, claude code 等のローカルファイルを読み込めるコーディング**AIエージェントツールであれば何でも**動作します

© 2026 NTT DATA Mathematical Systems Inc.

本日の内容

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

    1. AIによるシミュレータ開発

    2. AI開発のその先

        AIによるシミュレーションの活用・運用

        AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるAI活用

① モデルの構築・更新の負担

データ収集・現場ロジックの再現
（大量のif-thenルール）

→ **AIによるシミュレータ開発**
（AI develops simulator）

② 実験・結果の解釈の負担

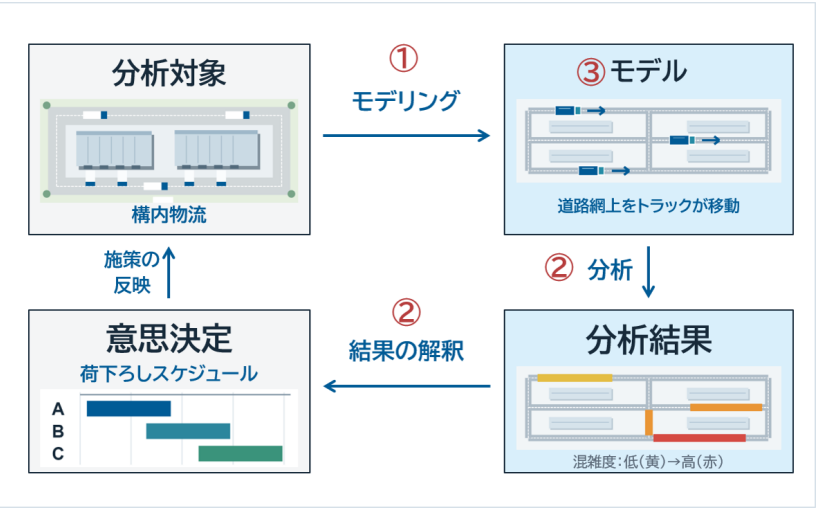
有益な示唆を得るには多くの
試行錯誤が必要

→ **AIによるシミュレーション活用・運用**
（AI operates simulation）

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
現実世界の要素が存在

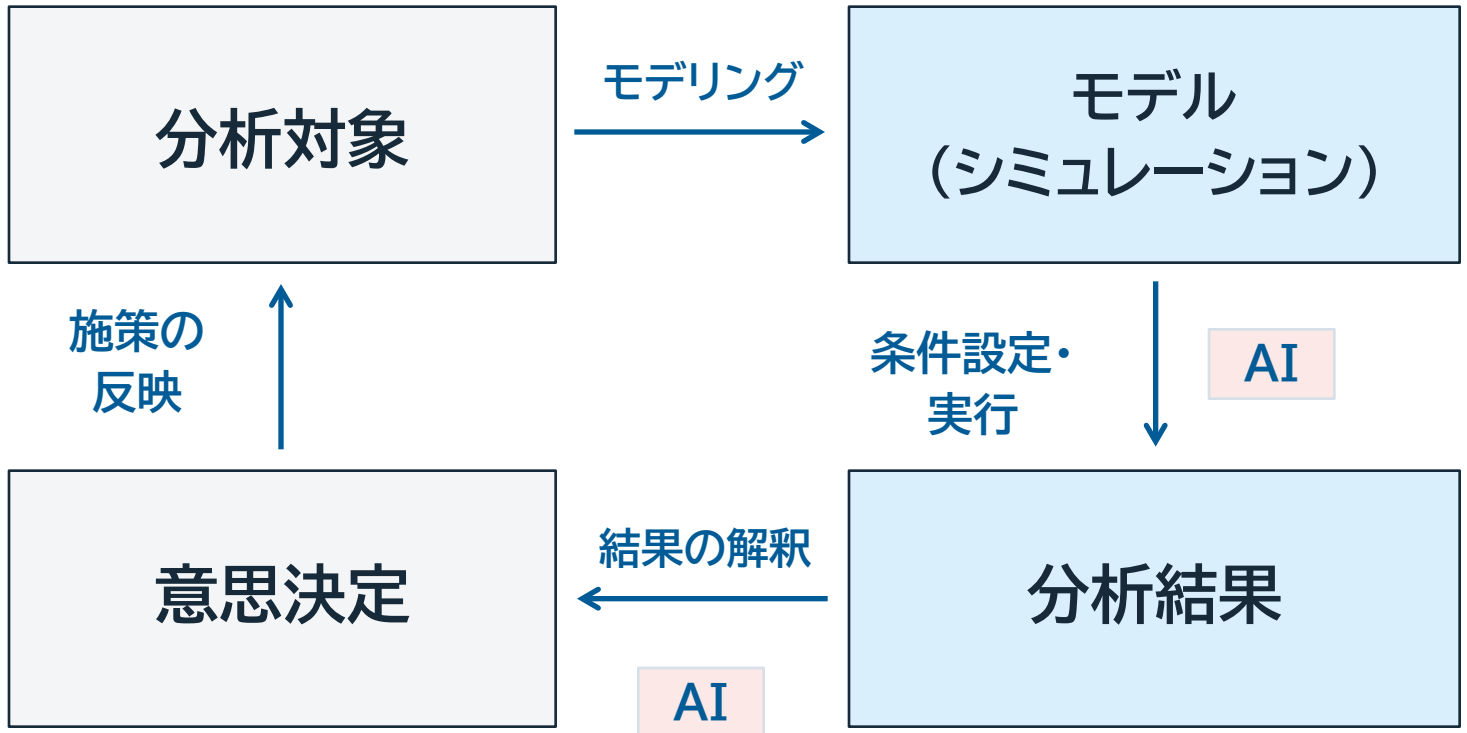
→ **AIを内包するシミュレーション**
（simulator includes AI）



© 2026 NTT DATA Mathematical Systems Inc.

AIによるシミュレーション活用・運用

構築したモデルの活用・運用をAIが行う



① 条件設定・実行

what-if分析／最適化を
ツールで繰り返す。
パラメータチューニングの自
動化も可能

② 結果の解釈

分析結果を比較し、
理由と候補案を整理する。

© 2026 NTT DATA Mathematical Systems Inc.

AIによるシミュレーション活用・運用

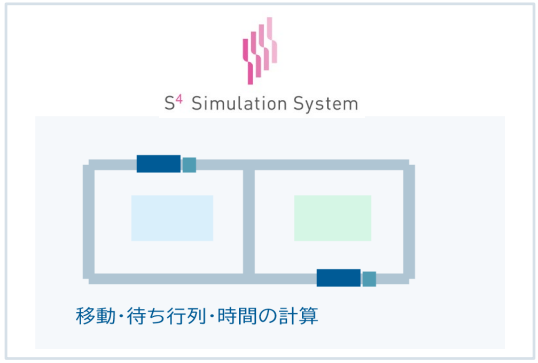
構内物流シミュレーションにおける活用イメージ

① 条件設定



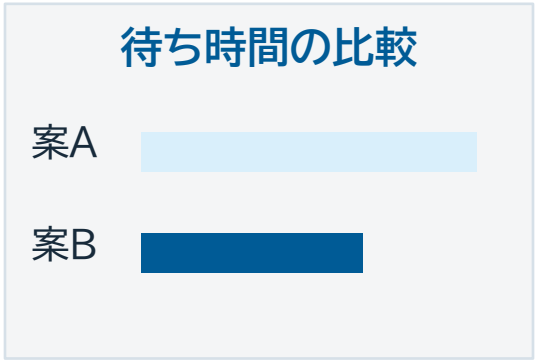
入構時刻を調整する。
例：到着の集中を分散

② 実行



同じモデルを使い、
変更した条件で計算

③ 分析



待ち時間・混雑を比較

④ 次の条件設定

条件を変えて
再試行

または

候補案を人へ返す

目的を達成するまで繰り返す

© 2026 NTT DATA Mathematical Systems Inc.

本日の内容

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

    1. AIによるシミュレータ開発

    2. AI開発のその先

        AIによるシミュレーションの活用・運用

        AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるAI活用

① モデルの構築・更新の負担

データ収集・現場ロジックの再現
（大量のif-thenルール）

→ **AIによるシミュレータ開発**
（AI develops simulator）

② 実験・結果の解釈の負担

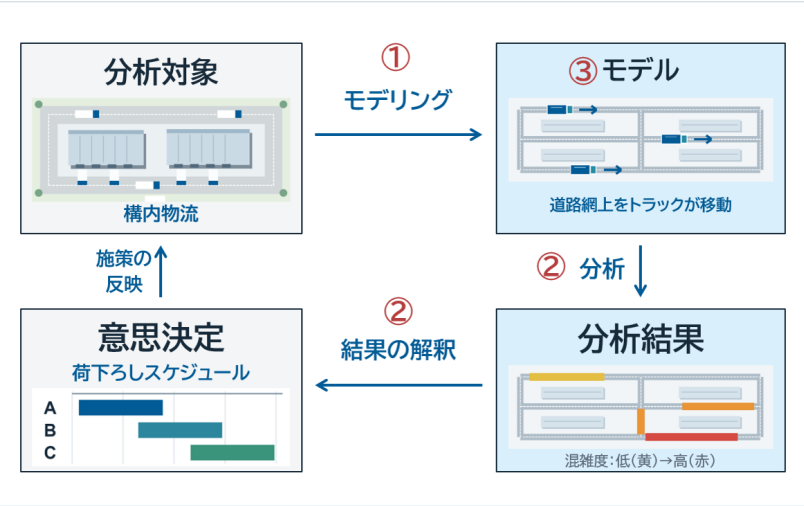
有益な示唆を得るには多くの
試行錯誤が必要

→ **AIによるシミュレーション活用・運用**
（AI operates simulation）

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
現実世界の要素が存在

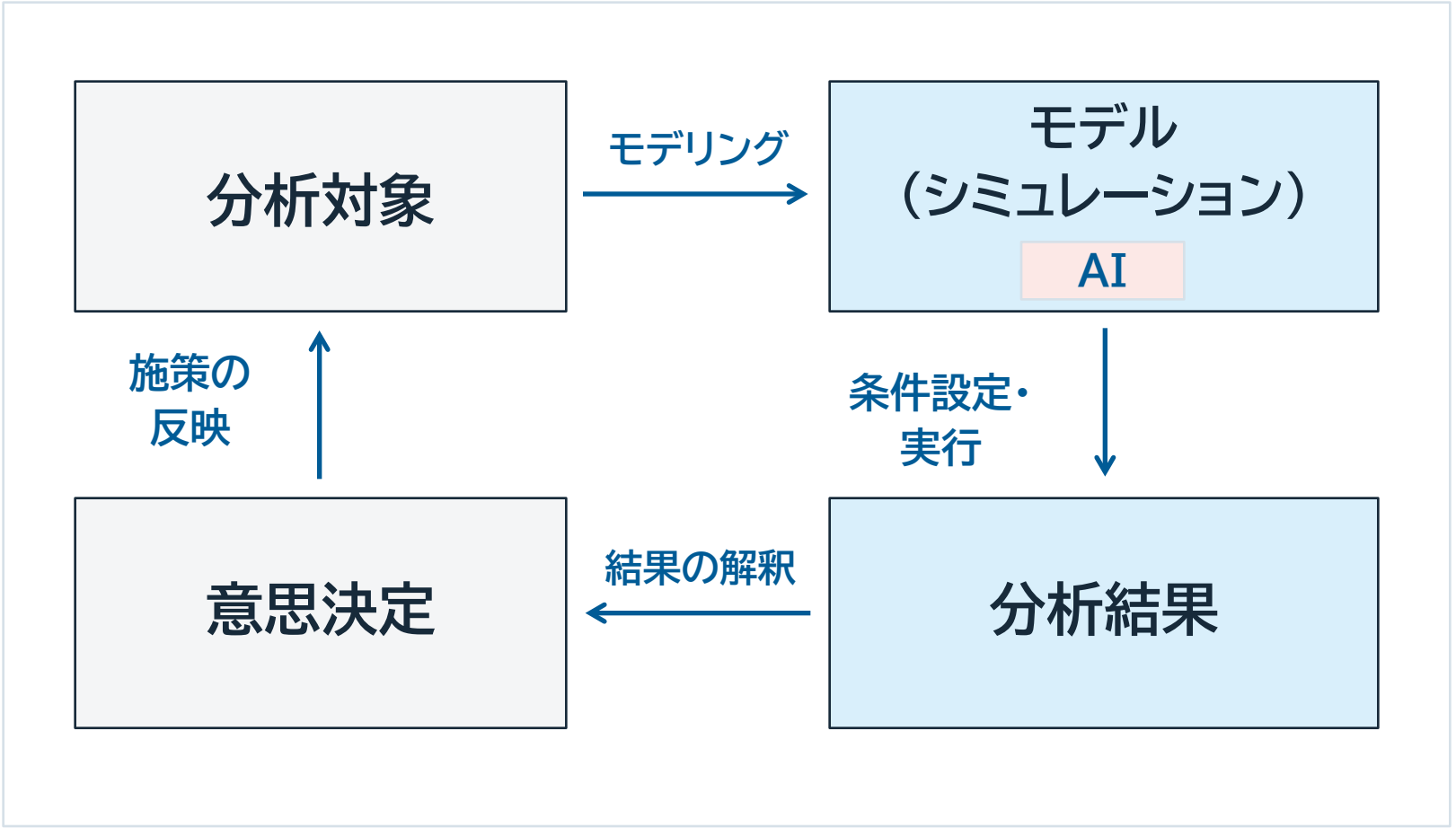
→ **AIを内包するシミュレーション**
（simulator includes AI）



© 2026 NTT DATA Mathematical Systems Inc.

AIを内包するシミュレーション

ルール化が難しいロジックをAIで表現



© 2026 NTT DATA Mathematical Systems Inc.

ルール化が難しいロジックの例

異常時：対応の判断



高度な知識に基づく判断が必要

人流：入店の判断



SNS：拡散の判断



人間の行動はルール化が困難
（無理やりルール化すると意味のないシミュレーションになりがち）

© 2026 NTT DATA Mathematical Systems Inc.

AIによるロジックの実装方針

① データドリブンなAI



模倣学習：
過去ケースの対応を学ぶ。
強化学習：
結果が良くなる方策を学ぶ。
→ **複雑な行動をデータから学習**

② LLMによる表現



状況・役割・目的を言葉で与え、
判断・行動を決めさせる
→ **人間の行動の曖昧さを**
LLMに押し付ける

③ 人による シミュレーションへの介入



異常時の対応を自然言語で指示。
→ **LLMが解釈して、**
実行中の動作に反映

© 2026 NTT DATA Mathematical Systems Inc.

本日の内容

1. デジタルツインにおけるシミュレーション技術の活用

2. デジタルツイン領域における生成AIの活用可能性

    1. AIによるシミュレータ開発

    2. AI開発のその先

        AIによるシミュレーションの活用・運用

        AIを内包するシミュレーション

3. まとめ

© 2026 NTT DATA Mathematical Systems Inc.

デジタルツインにおけるAI活用

① AIコーディングは強い
② ツール整備が重要

① モデルの構築・更新の負担

データ収集・現場ロジックの再現
（大量のif-thenルール）

→ **AIによるシミュレータ開発**
（AI develops simulator）

② 実験・結果の解釈の負担

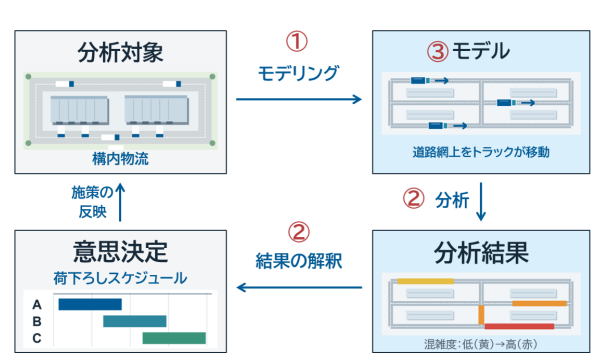
有益な示唆を得るには多くの
試行錯誤が必要

→ **AIによるシミュレーション活用・運用**
（AI operates simulation）

③ モデルの表現可能性の限界

ルールベースでは表現が難しい
現実世界の要素が存在

→ **AIを内包するシミュレーション**
（simulator includes AI）



© 2026 NTT DATA Mathematical Systems Inc.

生成AIによるシミュレーション構築

従来

構築コストが大きい
→ 失敗しないテーマを選んでから作る

候補を検討

有望な課題に絞り込み

要件定義・仕様確定

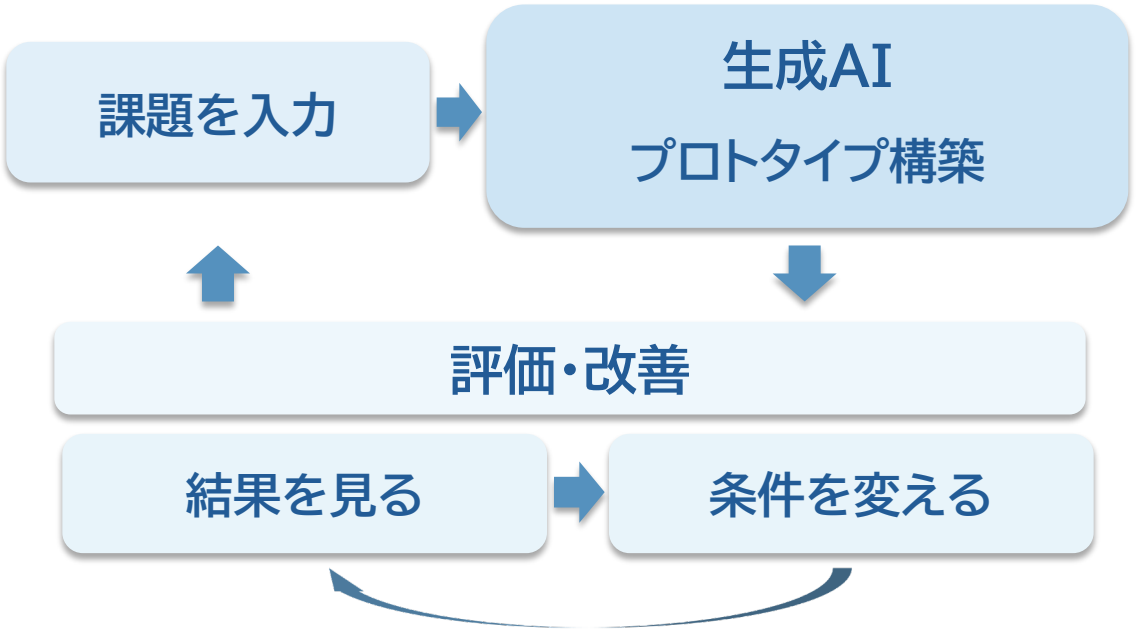
シミュレータ構築

評価・改善

入念な選定・要件定義 → 構築

生成AIのある世界

プロトタイプ構築のコストが小さい
→ 作ってから使えそうか考える



「何に使えるか」を探索

© 2026 NTT DATA Mathematical Systems Inc.


NTT DATA