

# 手戻りを含むプロジェクトのシミュレーション モデルに関する研究

## Proposing a Project Simulation Model Considering Rework

稗方 和夫<sup>1</sup> 満行 泰河<sup>2</sup> 王 汝佳<sup>1</sup>

Kazuo Hiekata<sup>1</sup>, Taiga Mitsuyuki<sup>2</sup>, and Rujia Wang<sup>1</sup>

<sup>1</sup> 東京大学大学院 新領域創成科学研究科

<sup>1</sup> Graduate School of Frontier Sciences, The University of Tokyo

<sup>2</sup> 横浜国立大学大学院 工学研究院

<sup>2</sup> Faculty of Engineering, Yokohama National University

**Abstract:** A project simulation model considering rework is proposed in this paper. The proposed model can be applied in duration estimation of large and complex projects based on the input extracted from standardized past log data. Tasks and their interdependency, along with the information of available human resources is inputted first, while delay and rework possibility will be extracted. As an output of the proposed model, predicted duration of the simulated project will be calculated, supporting decision making when starting a new project.

## 1 緒言

大規模複雑プロジェクトにおいて、効率よくスケジュールを管理することは非常に重要であり、モデルに基づいた作業現場のシミュレーションの運用に関する研究も長年に渡って研究されている。船舶の建造、都市基盤や工場の建設、およびスケールの大きい情報システムの開発など、様々な業界の大規模プロジェクトにおいて、作業進行途中の仕様変更による手戻りや、人員の経験不足などによる納期の遅延は、大きな経済的損失に繋がるが多いため、不確実性を含めたシミュレーションモデル[1]も提案されている。

しかし、既存のモデルを運用するためには、詳細かつ膨大な現場情報から不確実性を数値化する必要がある。この手順では、殆どの場合、多種多様な情報を人間系で処理しなければならない。結果として、プロジェクトの工期見積もりや工程管理において、専門家の経験と直感に頼るケースが多い[2]。

本研究では、手戻りモデルを含む工程シミュレーションモデル、および遅延と手戻りをシミュレートするための不確実性イベントの発生確率を、プロジェクトの実績データから算出する仕組みを提案し、大規模プロジェクトの工期見積もりにおける意思決定を支援する。

## 2 シミュレーションモデル

本節では、提案するシミュレーションモデルの構成、およびシミュレーションの手順について説明する。シミュレーションモデルは、先行研究[1]を参照して作成した基本モデル、作業の遅延を表す遅延モデル、および作業間の手戻りを表現できる手戻りモデルから構成され、資源制約付きプロジェクトスケジューリング問題（RCPPSP, Resource Constrained Project Scheduling Problem）として扱い、離散イベントシミュレーションを用いて、見積もり工数を出力する。

### 2.1 基本モデル

基本モデルは、タスクとタスク間依存関係、およびリソースチームから構成する。ここで、タスクとは大規模プロジェクトの主な作業手順であり、実行中、実際に記録される作業の単位でもある。リソースチームとは、プロジェクトの実行に携わる作業人員や設備を抽象化したものであり、リソースチームは作業能力の属性を持ち、各タスクに対するシミュレーション上のタイムステップ毎の進行工数を表す。また、タスク間依存関係の定義として、Finish-to-Startの依存関係のみとし、表 1 のように、DSM (Dependency Structure Matrix) で表現することができる。表 1 では、タスク B がタスク A に依存するこ

とを示している。

表 1. タスクと依存関係の例

Task	A	B
A		0
B	1	

各タスクとリソースチームに ID が付与されており、式(1)と式(2)のように表現する。なお、タスク $T_i$ が依存しているタスクの集合は、式(3)のように表現する。タスク $T_i$ は、集合 $PT_i$ の中の全てのタスクが完了するまで、着手してはいけない。

$$T_i, i = 1, 2, \dots, I \quad (1)$$

$$R_j, j = 1, 2, \dots, J \quad (2)$$

$$PT_i \subseteq \{T_i | i = 1, 2, \dots, I\} \quad (3)$$

## 2.2 遅延モデル

本研究のシミュレーション対象とする遅延は、作業員の作業能力の低減や設備トラブルによる作業の延長を含めず、1つのタスクを完了できる最小の工数に対して、実工数が大きくなることを意味する。また、本モデルでは、手戻りが考慮されており、同じ ID を持つタスクを複数回実行する場合がある。手戻りによるあるタスクの $o$ 回目の実行時には、式(4)のように、タスクの実行回数ごとに、最小の工数が $mwa_{i,o}$ として定義される。次に、1つのタスクに対して、式(5)のような遅延情報が複数定義される。ここで、 $dp_{i,o,n}$ とは、タスク $T_i$ が $o$ 回目実行され、工数 $n$ の遅延が発生する確率である。

$$mwa_{i,o} > 0 \quad (4)$$

$$dp_{i,o,n} \in (0,1] \quad (5)$$

遅延モデルに踏まえ、タスクの進捗率が式(6)のように定義される。ここで、 $pro_{i,o}(t)$ と $awa_{i,o}(t)$ は、それぞれ、タスク $T_i$ が $o$ 回目実行され、シミュレーション上の時刻 $t$ における進捗率と、実際に実行した工数である。また、タスクの完了は、残り工数 $rwa_{i,o}(t) < 0$ を用いて、判断する。ただし、 $rwa_{i,o}(t)$ は、タスクが $o$ 回目実行される前には、 $rwa_{i,o}(t) = mwa_{i,o}$ とする。

$$pro_{i,o}(t) = awa_{i,o}(t)/mwa_{i,o} \quad (6)$$

シミュレーション上の挙動として、タスク $T_i$ の進捗率が1に達した時点で、確率によって、 $n$ 工数の遅延が発生すると、 $rwa_{i,o}(t) \leftarrow rwa_{i,o}(t) + n$ の処理を行う。

## 2.3 手戻りモデル

本研究のシミュレーション対象とする手戻りとは、1つのタスクに問題が発生し、中断され、完成済みもしくは作業中のタスクに対する修正、あるいはやり直しが余儀なくされることである。1つのタスクに対して、式(7)のような手戻り情報が複数定義される。ここで、 $rp_{i1,o,pro,i2}$ とは、タスク $T_{i1}$ が $o$ 回目実行され、進捗率が $pro$ の時に、問題が発生し、手戻り先である、タスク $T_{i2}$ から手戻りする確率である。

$$rp_{i1,o,pro,i2} \in (0,1] \quad (7)$$

シミュレーション上の挙動として、手戻り発生時、タスク $T_{i1}$ の実際工数 $awa_{i,o}(t)$ と進捗率 $pro_{i,o}(t)$ が0に戻る。また、タスク $T_{i2}$ の残り工数は、時刻 $t$ における $T_{i2}$ の実行回数 $o_{i2}$ に応じて、 $rwa_{i2,o_{i2}}(t) = mwa_{i2,o_{i2}}$ として設定する。

## 2.4 離散イベントシミュレーション

離散イベントシミュレーションの大まかな流れは、先行研究[1]を参照し、遅延モデルと手戻りモデルに応じて、変更を加えたものである。シミュレーションのフローは、図1で示す。具体的な手順は以下の通り：

1. 初期化：入力ファイルから情報を読み込み、タスクとリソースチームの状態を初期化する。
2. 完了していないタスクに対して、クリティカルパスを計算する。
3. タスクとリソースチームに対して、それぞれ TSLACK（クリティカルパス上の近いタスクを優先する）と SSP の優先度ルール（各タスクに対する作業能力の総和の低いリソースチームを優先する）[1]に従い、タスクの割り振りを行う。
4. 実行可能なタスクの残り工数から、タスクを実行しているリソースチームの該当タスクに対する作業能力値の総和を引く。
5. タスクの手戻り情報から手戻りの発生確率を読み込み、手戻りの発生を判断する。
6. 手戻りが発生した場合、2.3 で述べた一連の処理を行う。
7. 進捗率が1に達したタスクに対して、遅延の発生を判断する。

8. 遅延が発生した場合、2.2 で述べた一連の処理を行う。

9. タスクとリソースチームの状態を更新する。

10. タスクの完了を判断する。全てのタスクが完了した場合、シミュレーションを終了し、ガントチャートを出力する。終了していないタスクがある場合、時刻  $t \leftarrow t + 1$  し、2. クリティカルパスの計算に戻る。

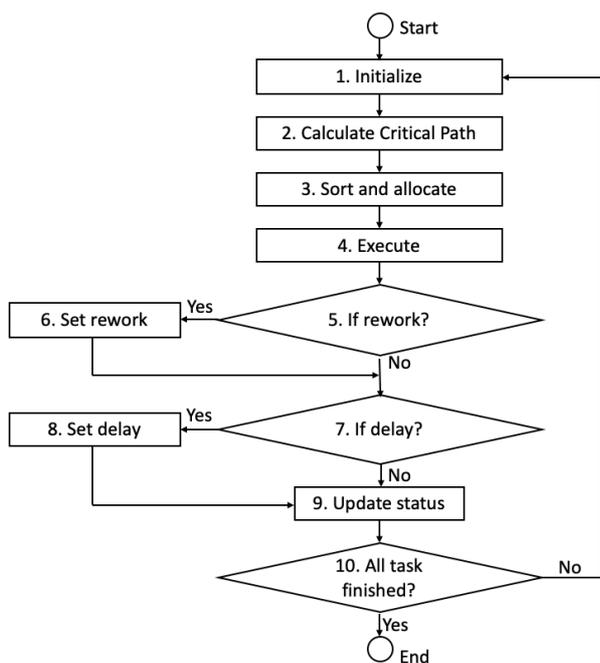


図 1. シミュレーションのフロー

### 3 実績データの抽出

本節では、2.2 と 2.3 で述べた手戻りモデルと遅延モデルを、一定形式の実績データから抽出する方法について紹介する。

#### 3.1 実績データの形式

本研究では、式(8)のように、プロジェクト、作業単位としてのタスク、開始時間、完了时间及びタスクを実行したリソースの進捗能力を含めたログデータを 1 件のログデータとし、複数件のログデータの集合を実績データとする。これらの情報は、大規模プロジェクトにおいて、最低限記録されるものと想定する。ただし、進行中のタスクにおいて、完了と判断されるまでに、完了時間を記録しなかったものとする。つまり、手戻りによる作業の中断は、記録されなかったことを仮定する。なお、タスク間の依存関係も既知情報とする。

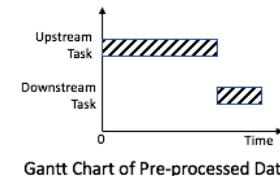
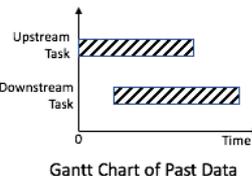
$$\{\text{Project Name, Task Name, Start Time, End Time, Resource Capacity}\} \quad (8)$$

#### 3.2 実績データの前処理

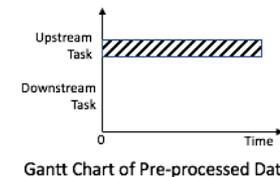
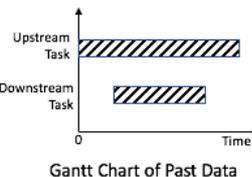
実作業において、Finish-to-Start の依存関係を厳密に従わない場合が多く、また、ログデータで誤りが存在する可能性もあるため、実績データに対する前処理が必要である。この段階で、Finish-to-Start の依存関係と矛盾しているデータを、依存関係で上位にあるタスクを優先して、処理する。つまり、依存関係で下位にあるタスクが実行している途中に、上位タスクも実行されているログデータがある場合、下位タスクが中断、遅れて開始、または実行されていないと見なす。

具体的に、1つのプロジェクトのログデータにおいて、異なるタスクのデータを2つずつ比較し、以下の手順を行う。各手順の概略は、図2に示す。ここで、依存関係で上位にあるタスクの開始、完了時間を  $st_{up}$ 、 $et_{up}$  とし、下位にあるタスクの開始、完了時間を  $st_{down}$ 、 $et_{down}$  とする。

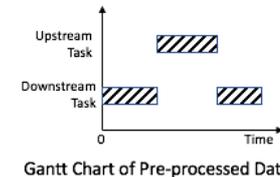
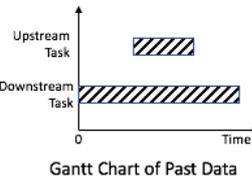
Situation 1



Situation 2



Situation 3



Situation 4

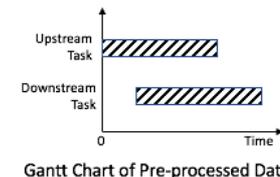
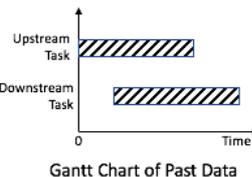


図 2. 前処理の概略図

$et_{up} > st_{down}$ であれば、依存関係に反したデータとして認識する。実行時間の重ね、つまり $et_{down} > st_{up}$ も満たしていれば、以下の4つの場合に応じて、適切な処理を行う。

1.  $st_{up} < st_{down}$ かつ $et_{up} < st_{down}$ の場合： $st_{down} \leftarrow et_{up}$ とする。
  2.  $st_{up} < st_{down}$ かつ $et_{up} > et_{down}$ の場合：下位タスクのログデータを消去する。
  3.  $st_{up} > st_{down}$ かつ $et_{up} < et_{down}$ の場合：開始時間が $et_{up}$ 、完了時間 $st_{down}$ 、他情報が下位タスクのログデータと同様なログデータを一件構造し、 $st_{down} \leftarrow et_{up}$ とする。
  2.  $st_{up} > st_{down}$ かつ $et_{up} > st_{up}$ の場合： $et_{down} \leftarrow st_{up}$ とする。
- 1回のデータ処理を行うと、データに変更があるため、もう一度最初から比較を行う。前処理は、全てのデータに変更がなくなるまで、繰り返して行なわれる。

### 3.3 遅延情報の抽出

前処理の次に、最小の工数と遅延確率から構成する遅延モデルの情報を抽出する。

まず、ログデータ一件に対して、当時のプロジェクト内における実際工数、実行回数と実行順番を記録する。実際工数は、実行時間とリソースの進捗能力の乗積である。実行回数は、1つのプロジェクトにおける同じタスクのログデータの開始時間に基づき、1から順につける。また、実行順番とは、1つのプロジェクトにおいて、ログデータごとに、次に実行したタスク名の記録である。ただし、実行順番は、手戻り先を抽出するための記録であるため、次回タスクの開始時間は、当タスクの完了時間より早いことはない。

次に、最小の工数を抽出する。最小の工数は、実行回数ごとに定義されるため、プロジェクト構わず、タスク名と実行回数が同様であるログデータに対して、実際工数の最小値を求める。この最小値を、該当タスクの最小の工数とする。ここで、ログデータ一件において、次回タスクが依存関係における上位タスクの場合、手戻りが発生したと見られ、作業が中断したと意味するため、この計算に含めない。

最後に、遅延確率を求める。ログデータ一件に対して、遅延工数を式(9)のように定義する。ここで、 $dwa$ 、 $ad$ 、 $i$ と $o$ は、それぞれ、ログデータにおける遅延工数、実際工数、タスク名、および実行回数を意味し、簡略化のため、 $dwa \in \mathbb{N}$ とする。手戻りの発生によって、 $dwa < 0$ になった場合、 $dwa = 0$ とする。

遅延確率の計算は、式(10)のように求める。ここで、 $count_{i,o,n}$ とは、タスク名が $i$ 、実行回数が $o$ 、遅延工数

が $n$ のログデータの件数である。

$$dwa = \text{ceil}(ad/mwa_{i,o}) \quad (9)$$

$$dp_{i,o,n} = count_{i,o,n}/count_{i,o,0} \quad (10)$$

### 3.4 手戻り情報の抽出

3.2で言及したように、手戻りの発生は次回タスクから判断する。手戻り発生時の作業進捗率 $pro$ は、0.1を進捗単位とし、 $pro \in \{0.1, 0.2, \dots\}$ とする。実績データ内の進捗率とは、ログデータの完了時進捗率であり、式(11)のように定義する。ただし、遅延の発生によって、進捗率が1を超える場合もある。

$$pro = \text{floor}(ad \times 10/mwa_{i,o})/10 \quad (11)$$

ログデータ全体に対して、式(12)のように、手戻り確率の集計を行う。ここで、 $count_{i,o,pro,j}$ とは、タスク名が $i$ 、実行回数が $o$ 、完了時の進捗率が $pro$ 、手戻り先がタスク $T_j$ のログデータの件数であり、 $count_{i,o,pro}$ とは、タスク名が $i$ 、実行回数が $o$ 、完了時の進捗率 $\geq pro$ のログデータの件数である。

$$rp_{i,o,pro,j} = count_{i,o,pro,j}/count_{i,o,pro} \quad (12)$$

### 3.5 シミュレーションへの適応について

実績データの情報が限られているため、抽出した手戻り情報は、実作業のシミュレーションと必ず一致しているとは言い難い。したがって、シミュレーション上のタスク実行回数は、記録における最大実行回数を超える可能性がある。この場合、取得した最大実行回数の最小の工数を使用する。

## 4 ケーススタディ

ケーススタディでは、仮想的なプロジェクトのデータを用いて、提案した一連の仕組みを検証する。

データから、遅延と手戻りモデルの情報の正解データを作成し、プログラムから得た情報と比較し、一致することによって、情報抽出のアルゴリズムを検証する。シミュレーションの正常なる作動は、シミュレーション結果で、設定された遅延と手戻りの発生があることで検証する。情報抽出のアルゴリズムと、シミュレーションモデルの一致性は、シミュレーション結果から、情報抽出を行い、モデルの数値と比較することで検証する。

#### 4.1 実績データの構成

実績データは、プロジェクトの一部を想定し、依存関係を持つ2つのタスクが100回繰り返し、この2つのタスクに以下の5つのパターンが発生したと想定し、タスク間依存関係は、表1の設定を使用する。リソースについて、全ログデータで、1日毎に1工数のリソース進捗能力があったと想定する。すなわち、全ログデータのResource Capacityを1にする。実績データの各パターンのガントチャートと工期の分布は、図3と図4に示す。パターン1~5の繰り返し回数は、それぞれ40、18、2、10、30回とする。

補足として、ガントチャート内の黄色の部分、実績データの前処理で除去される部分である。

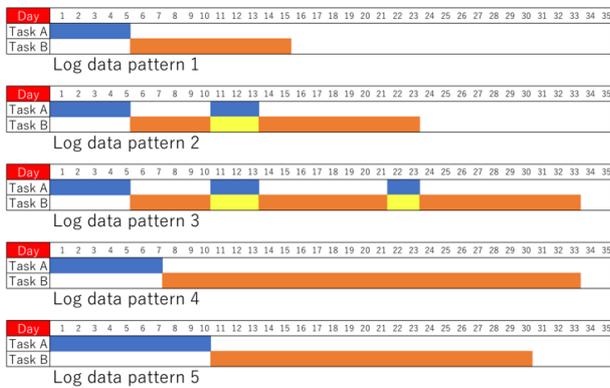


図3. 実績データのガントチャート

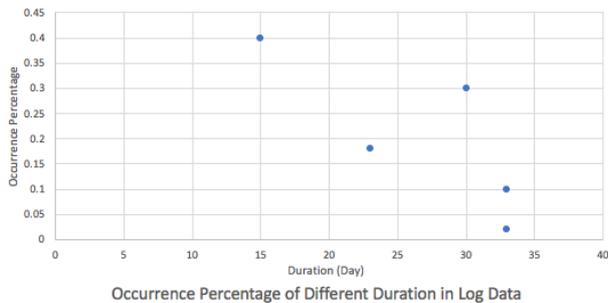


図4. 実績データの工期分布

#### 4.2 抽出した遅延モデルと手戻りモデル

プログラムによる計算結果は表2に示す。この結果が、データから作成した正解データと一致していることによって、モデルのパラメータの抽出が検証された。

表2. 抽出した遅延モデルと手戻りモデル

	Delay Model	Rework Model
Task A	$mwa_{A,1} = 5$ $mwa_{A,2} = 3$ $mwa_{A,3} = 2$ $dp_{A,1,2} = 0.1$ $dp_{A,2,5} = 0.3$	-
Task B	$mwa_{B,1} = 10$ $mwa_{B,2} = 10$ $mwa_{B,3} = 10$ $dp_{B,1,10} = 0.3$ $dp_{B,1,16} = 0.1$	$rp_{B,1,0.5,A} = 0.2$ $rp_{B,2,0.8,A} = 0.1$

#### 4.3 シミュレーション結果

同様なタスク、タスク間依存関係、および表3で示すリソースチームの情報から構成する基本モデルを入力し、4.2の抽出結果を加え、シミュレーションを1000回実行した結果は図5に示す。

表3. リソースチームの設定値 (工数/日)

Task ID	A	B
Resource Team 1	1	0
Resource Team 2	0	1

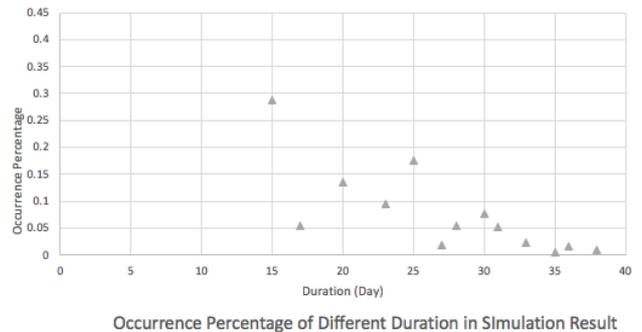


図5. 1000回シミュレーションの結果

シミュレーションの結果、提案したモデルは、各タスクにおける遅延および手戻りの再現ができたことを示すことができた。

ここで、いくつかの代表的なシミュレーション結果を、図6で示す。結果1と結果2は、タスクAでの遅延のみを表現した。結果3と結果4は、タスクAの遅延、およびタスクBでの手戻りを表現した。結果5と結果6は、タスクBでの遅延のみを表現した。結果7と結果8は、タスクBでの手戻りのみを表現した。結果9と結果10は、タスクAとタスクBの遅延を表現した。

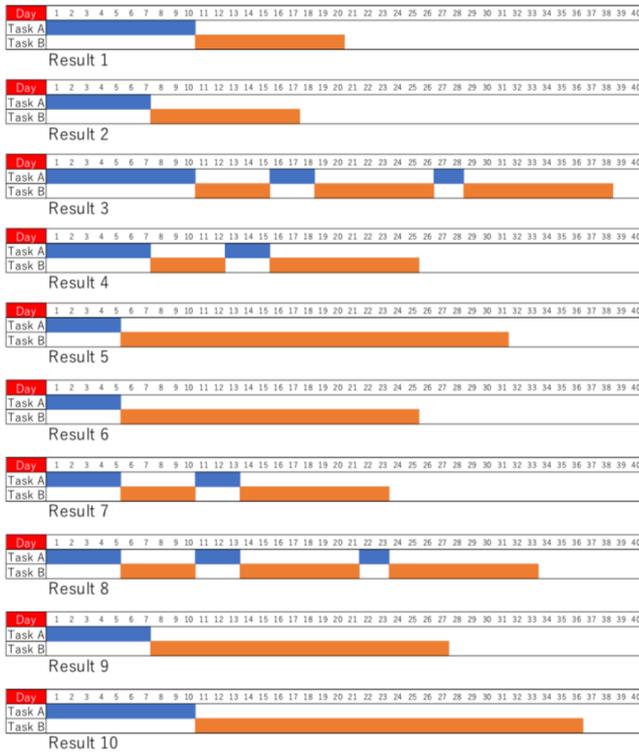


図 6. 代表的なシミュレーション結果

シミュレーションの結果から、情報抽出を再度行った遅延および手戻りモデルのパラメータは表 4 に示す。

表 4. シミュレーション結果からの逆算結果

	Delay Model	Rework Model
Task A	$mwa_{A,1} = 5$ $mwa_{A,2} = 3$ $mwa_{A,3} = 2$ $dp_{A,1,2} = 0.109$ $dp_{A,2,5} = 0.292$	-
Task B	$mwa_{B,1} = 10$ $mwa_{B,2} = 10$ $mwa_{B,3} = 10$ $dp_{B,1,10} = 0.246$ $dp_{B,1,16} = 0.075$	$rp_{B,1,0.5,A} = 0.203$ $rp_{B,2,0.8,A} = 0.142$

## 5 考察

提案手法では、詳細な情報を持つシミュレーションモデルを提案した。実行回数や進捗率を含めた、不確実性イベントの発生確率が入力され、実際の工程に対して、良い再現性があると思われる。また、過去にある実績データの多少を構わず、提案した仕組みを適応することができる。一方、シミュレーシ

ョンモデルにおける大量な入力情報は、実績データから自動的に抽出する手法も提案したため、煩雑かつ主観的な作業を軽減することができる。不確実性に纏わる確率の数値は、過去に発生したもののみを捉えたため、シミュレーションに不必要な情報が含まれない。

しかし、提案したモデルは一般的であるため、タスクとリソースチームの粒度、および依存関係などは、事前に人間系で決める必要がある。

今後の研究では、実際のプロジェクトにおいて、獲得できる情報を調査する。それらの情報に対する利用の最大化、およびモデルに対する再検討を行うことによって、より正確な工期見積もり方法を提案する。

## 6 結論

本研究では、手戻りを含めたシミュレーションモデル、および実績データから遅延と手戻りモデルを抽出する仕組みを用いて、精度の高い工期見積もり手法を提案した。過去の実績における遅延や手戻りを、シミュレーション上で再現することで、工程管理の意思決定を支援できると考えられる。

## 参考文献

- [1] 満行泰河, 稗方和夫, 松原洗也: 船舶建造プロセスシミュレーションを用いた生産設備の導入に関する研究, 日本船舶海洋工学会論文集, Vol. 24, pp. 291-298, (2016)
- [2] Minhoe Hur, Seung-kyung Lee, and Bongseok Kim: A study on the man-hour prediction system for shipbuilding, Journal of Intelligent Manufacturing, Vol. 26, No. 6, pp. 1267-1279, (2015)