「マルチメディア，分散，協調とモバイル
(DICOMO2017)シンポジウム」 平成29年6月

生産現場における作業監視のためのイベント遷移パターンの設計手法

中田 侑[1]　木下 雅文[1]　宮田　辰彦[1]

概要：近年 IoT 技術が注目され、製造分野でも生産現場の作業監視等で適用されている。しかし、センサー等の設置やネットワークのシステム構築コストが大きくなる問題があった。そこで、工作機械等の設備が保守用に出力する、設備の内部動作（イベント）が記載されたイベントログから、人の作業を捉える手法を検討した。これまで、イベントと作業の対応関係の定義と、イベントから作業を特定するプログラムの設計に時間を要し、開発コストが高くなる問題があった。これに対し本研究では、イベントログと作業の対応関係を示すモデルと、そのモデルに基づく設計開発手法を提案し、プログラムの作成期間を 1/20 に抑えらることを確認した。これにより、イベントログから作業を捉えるための人件費を抑え、低コストな作業監視の実現に貢献する。

Design Technique of Event Transitional Patterns to Reveal Operations in Manufacturing Field

YU NAKATA[1]　MASAFUMI KINOSHITA[1]　TATSUHIKO MIYATA[1]

1. はじめに

近年 IoT 技術が注目され、製造業においては、生産現場における人の作業監視などに適用されている [1]。作業監視を行うためには、生産現場にセンサー等を設置し、作業時の環境情報等のデータを取得・加工することで、生産性の見える化や生産改善等を行う。作業監視に必要なデータを収集するために、生産現場にセンサーやカメラを新たに設置し、温度や音、光などの環境情報や、人や製造物、設備の動きの動画像を取得する [2,3]。そして、それらのデバイスから得られたデータを統合・加工し、作業監視を行い、生産性の見える化や改善施策・生産計画の立案に利用する。

しかしながら、作業監視用のシステムの構築には、多額のコストを要する。高額なコストを要する要因は、主に現場へのセンサーやカメラの敷設コストやネットワークの敷設コストである。生産現場の環境は、工場ごとに異なっているため、顧客環境ごとに敷設計画や動作検証が必要となる。

そこで、生産現場に設置された設備が出力するイベントログから、人の作業を捉える手法を検討した。イベントログとは、設備が機械メーカーのメンテナンス用に出力するログであり、設備の内部動作（以降、イベントと呼ぶ）が、発生時刻と共に、時系列に出力されている。イベントログに記載された過去のイベントから、そのイベント発生した時の “加工中”、“材料待ち”、“段取り替え中”、“手作業中” といった工程内の作業を推測することで、設備投資を不要として、生産性分析が可能となる。

イベントログから、工程内の作業を推定するためには、イベントの並びと “自動加工中”、“段取り替え中” などの作業の対応関係を定義し、イベントログから作業を抽出する変換プログラムを開発する。現状では、この対応関係の定義と変換プログラムの設計開発に多くの時間を要し、開発コストが高くなる。そのため、イベントログの解析に人件費がかかり、作業監視を実施するためのコストが抑えられない。

そこで我々は、対応関係の定義と実装を簡潔化する方法として、イベントログと作業の対応関係を示すモデルと、そのモデルに基づく設計開発手法を提案する。本手法では、イベントの出現順序のパターンと、複数のイベントによる

[1] （株）日立製作所 研究開発グループ

一連の出現に対する工程の状態との対応関係をモデル化する。また、モデルの設計において、本技術は、過去に設備で発生したイベントログを入力として、設計者による、イベントの出現順序のパターンのモデル化を簡潔化する。さらに、モデル駆動型開発 [4] に基づき、設計者が設計したモデルを、実行コードに変換するコンパイラを開発した。これにより、イベントログと工程内の作業との対応関係の定義、仕様設計、実装を簡潔にし、開発コストを抑える。

本手法を、再度、金属旋盤工程の工作機械のイベントログに対して実行し、イベントログから工程の状態を把握するまでに要する期間を評価した。その結果、イベントと作業の対応関係の定義・実装に要した期間が 40 人日から 2 人日に短縮された。これにより、本技術は、低コストの生産改善 IoT ソリューションの実現に貢献できるものであることが示された。

本論文の構成は、以下の通りである。2 章では、3 章では、従来のアプローチにおける技術的問題および課題を説明し、本研究のアプローチを説明する。4 章では、本技術の効果を評価する。5 章では、本研究のまとめを説明する。

2. 生産現場における作業監視

2.1 作業監視の目的と課題

機械加工が行われる生産現場では、材料から製品の完成に至るまでに、複数の工程が存在する。各工程において作業員が工作機などの設備を用いて材料を加工し、加工品を生成する。各工程において、材料投入から加工品の完成に至るまでには、材料の投入や、設備の動作プログラムの設定、自動加工、手動操作による加工の補正など、様々な作業が含まれ、これら一連の作業が、製品の受注数だけ繰り返される。本論文では、材料投入から加工品の完成までの、繰り返される一連の作業を加工サイクルと呼び、一回の加工サイクルに要する時間をサイクルタイムと呼ぶ。

生産現場の生産性は、一定時間内に生産できる製品の数に依存するため、サイクルタイムの短時間化と平準化が求められている。しかしながら、従来よりサイクルタイムは、作業員が定期的にストップウォッチ等で計測しており、常に計測し続けることは困難であった。また、加工サイクル内のどのような作業で、作業員が時間を要しているのかまでは捉えることができず、サイクルタイムの平準化に向けた施策を立案することは困難であった。

そこで近年では、生産現場にセンサーやカメラを導入しサイクルタイムを計測し、さらに加工サイクル内の作業を細分化し、サイクルタイムを細分化された作業ごとに見せることで、サイクルタイムの平準化に向けた施策検討を目指すことが行われている [5, 6]。図 1 に、サイクルタイムを細分化して捉えるための一般的な方法を示す。生産現場でデータが収集された後、サイクルタイムと細分化された作業ごとの時間を表示するように、収集されたデータが変換される。最後に、データを可視化して、生産現場の作業改善に活かされる。

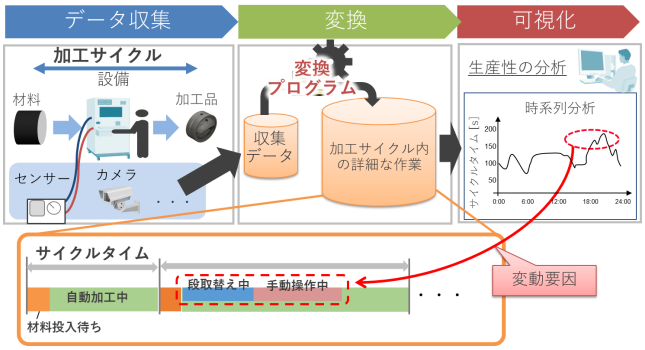


図 1 人の作業監視による生産性の分析

しかしながら、生産現場にセンサーやカメラを設置し、システムを統合することは、多くのコストを要する。高額なコストを要する要因は、主に現場へのセンサーやカメラの敷設コストと、それらで収集したデータを DB や分析システムに転送するためのネットワークの敷設コストである。さらに、生産現場の環境は、工場ごとに異なっているため、システム構築が顧客環境ごとに異なり、適切な敷設計画や動作検証が必要となる。

2.2 イベントログを用いた作業監視

本研究では、生産現場へのセンサーやカメラ等のデバイスの設置や、ネットワーク等のシステムの敷設を避けるため、既に現場に存在しているデータを二次利用し、作業監視を行う方法を検討した。そこで我々は、設備が出力するイベントログに着目した。

イベントログには、設備の内部動作（イベント）が、発生時刻や内部動作のパラメーターと共に、時系列に出力されている。イベントログに記載されたイベントから、“自動加工中”、“材料投入待ち”、“段取り替え中”、“手動操作中” といった、その設備を使用した作業を推測することで、設備投資を不要として作業監視が実現できる。

図 2 にイベントログの例と、イベントログから加工サイクル内の作業を捉える流れを示す。イベントログ内の特定のイベントから、その後に発生した特定のイベントまでを、一つの作業として捉える。その際には、イベントと作業の対応関係 (以降ではマッピングルールと呼ぶ) を定義し、その対応関係に基づきイベントログから作業を抽出するための “変換プログラム” を作成する。この変換プログラムに対して、イベントログを入力することで、各加工サイクル内で行われた作業とその流れを抽出する。

3. 提案手法

本章では、まず、この期間に行われる作業を整理し、時間を要する本質的な要因を示す。次に、提案手法として、イベントの出方と作業との対応関係を示すモデルを提案

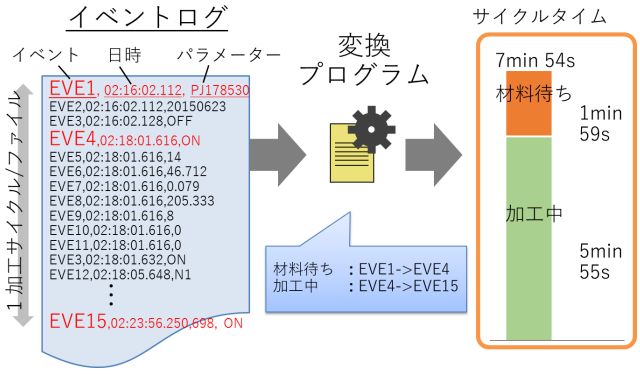


図 2　イベントログ

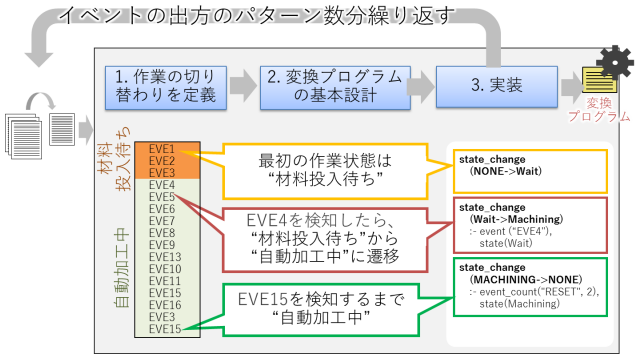


図 3　変換プログラムの作成における課題

し、さらにそのモデルを用いた迅速な設計開発手法を提案する。本報告では、提案するモデルをイベント遷移モデルと呼ぶ。本章で、イベント遷移モデルの概要と、それを用いた設計開発の手順を説明する。

3.1　技術課題

図 3 に、マッピングルールの検討と、変換プログラムの設計開発において、従来行われていた作業を示す。この作業では、主に以下の 3 つの手順を繰り返す。

(1) イベントの並びから、作業の区切りとなる箇所を特定
(2) イベントの並びから、各作業を捉えるための仕様を策定
(3) 策定した仕様を基に、変換プログラムを実装

1 では、変換プログラムの開発者が、各ファイルに含まれるイベントの並びを目視し、イベントの意味と、生産現場で起こりうる作業内容から、イベントの並びの中で作業の区切りとなる箇所を特定する。本研究の前提として、イベントログの各ファイルが、加工サイクルごとに分かれて出力されているものとする。そこで、イベントログの各ファイルの最初は、常にある特定の作業状態と見なすことができる。図 3 の例では、開発者は、イベント EVE1 が発生してからイベント EVE3 が出現するまでを材料投入待ちの状態 (作業) と見なし、イベント EVE4 が出現したことで自動加工中の作業に移行したとみなしている。その後、EVE4 から EVE15 まで、自動加工中の作業が続いていたと判断している。

2 では、1 で捉えた区切りに対し、イベントの列から各作業を捉えるための仕様を検討する。図 3 の場合、“最初の作業を材料投入待ち中とする”、“材料投入待ち中においてイベント EVE4 が発生したら自動加工中に遷移したとみなす”、“EVE15 が 2 回出現したら自動加工中が終了したとみなす” といったイベントと作業との対応関係を仕様として策定する。

3 では、2 で策定した仕様をプログラムに実装する。そして、加工サイクル内のイベントの出方をパターン化し、各パターンに対して、1 から 3 の流れを繰り返す。

この手順の中に、迅速にイベントログから作業を捉える上で、技術課題が 2 点存在する。

一点目は、2 において、全ファイル間で整合性のとれた仕様を策定することが困難な点である。あるファイルに対し、1 を行った後、別のパターン対しマッピングルールを定義する場合、既に策定したパターンに対するマッピングルールが変わらないようにしなければならない。そのため、過去に仕様を策定した全ファイルのイベントの並びを照合し、新しいファイルに対する仕様を策定しなければならない。しかし、ファイルのパターン数が多い場合や、ファイル内のイベントの数が大きいと、作業に多くの時間を要する。この課題を解決するためには、イベントの出方が異なる複数のパターンにおいて、統一的にイベントと作業の対応関係を定義できる設計手法が必要である。

二点目は、1 において、ファイルのパターン数やファイル内のイベント量が多い場合、作業の区切りとなる箇所を特定する際の作業量が大きくなる点である。この課題を解決するためには、パターン間でのイベントの出方の違いを迅速に発見し、その違いに相当するイベントのみに対して、作業との対応関係を定義する仕組みが必要である。これらの課題により、マッピングルールの定義と、変換プログラムの仕様作成・実装には多くの期間を要し、作業監視を実現するまでの人的コストが増大する。

3.2　イベント遷移モデルを用いた設計手法

3.1 節で説明した技術課題を解決するための基礎概念として、我々はまず、イベントの出方と作業との対応関係を示すモデルを提案する。本研究では、提案するモデルをイベント遷移モデルと呼ぶ。このモデルを基に、イベントの並びと生産現場での作業との対応関係の設計を簡潔化させるイベント遷移モデルの設計手法を提案する。これにより、イベントログにおけるファイルのパターン間で整合性のある設計を簡単化する。また、ファイルのパターン間で、イベントの並びにおける差異を抽出するアルゴリズムを提案し、抽出された差異の部分のみに対して開発者にイベントと作業との対応関係を規定させる設計手法を提案する。本研究では、パターン間でのイベントの並びを抽出する手法

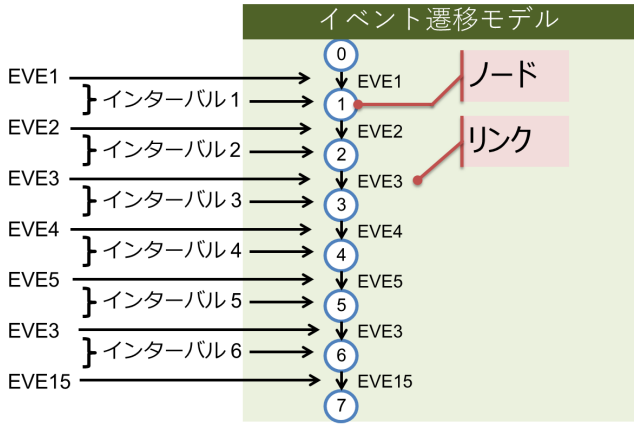


図 4　イベント遷移モデル

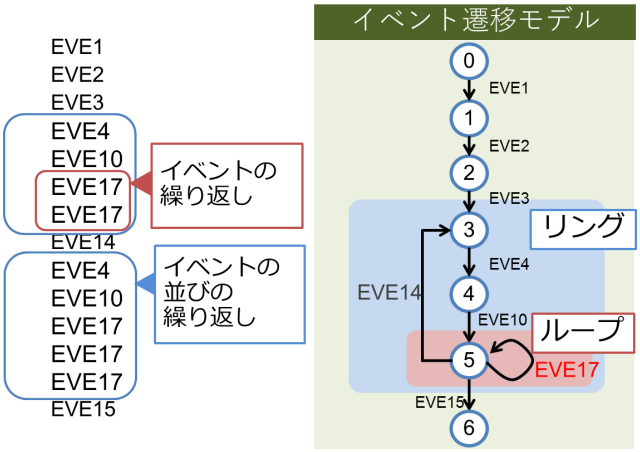


図 5　イベント遷移モデルによるイベントの繰り返し出現の記述

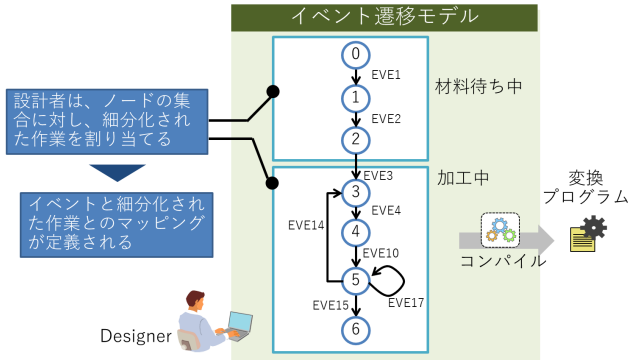


図 6　イベント遷移モデルによるイベントと細分化された作業とのマッピングの定義

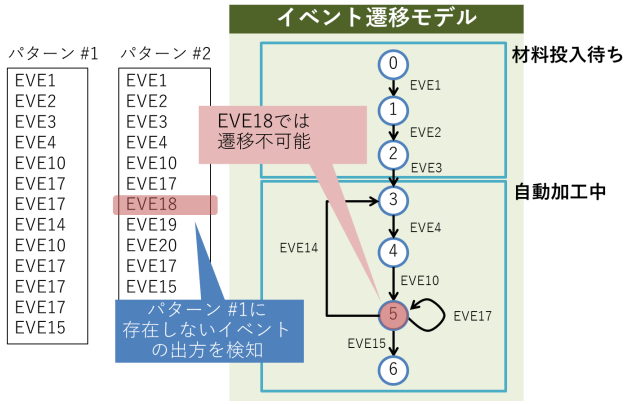


図 7　イベント経路追跡アルゴリズム

を、イベント経路追跡アルゴリズムと呼ぶ。これにより、多量のパターンが存在しても、迅速に作業の区切りとなる箇所を特定できるようになる。

以降では、イベント遷移モデルとイベント遷移モデルに基づく二つの設計アプローチを説明する。

3.2.1　イベント遷移モデル

イベント遷移モデルでは、イベントを有向辺リンクと定義し、あるイベントが出現してから次のイベントが出現するまでの間隔をノードとする (図 4)。また、イベントと作業との対応関係は、作業を意味する枠でノードを囲むことにより設計する。また、各ファイルの最初のイベントが出るまでを、ルートノード (図 4 のノード 0) として存在させ、最後のイベントが出た後には終端ノード (図 4 のノード 7) を設置する。開発者は、ルートノードから有向辺リンクをたどることで、イベントの出方を捉えることができる。

3.2.2　イベント遷移モデル設計手法

パターン間で整合性のある仕様を簡潔に設計するためには、複数のイベント出現パターンに対し、統一的にイベントと作業の対応関係を定義できる設計手法が必要である。これに対し、イベント遷移モデルでは、設計者が多様なイベントの出方を、簡潔な形式で設計することができる。例えば、生産現場で稼動する設備の場合、設備の中で特定の内部動作が繰り返されることが多い。その場合、同じイベントが複数回発生する。ただし、加工サイクル間において、繰り返される回数の違いを捉えることが特に重要でない場合は、繰り返しの回数を無視して、イベントの出方を捉える必要がある。そこでイベント遷移モデルでは、図 5 に示すように、特定のイベントの繰り返しをループとして、また特定の一連のイベントの繰り返しをリングとして設定することにより、これらのイベントの繰り返しの回数を無視し、繰り返しが起きうることのみを捉えた仕様を策定できる。

ノードは、ある一瞬の出来事を示すイベントと、その次のイベントまでの期間を示すため、各作業の期間は、ノードの集合として定義される。そこで、イベント遷移モデルにおいて、イベントと作業との対応関係は、図 6 に示すように、ノードを、加工サイクル内の各作業を意味する枠で囲むことにより設計する。

また、作成されたモデルは、ノードとリンク、枠の構造を自動的に判別し、変換プログラムを出力するコンパイラを用意している。このコンパイラにより、変換プログラムの開発期間も短縮することができる。

3.2.3　イベント経路追跡アルゴリズム

多量のパターン間において、イベントの出方から作業の

区切りを見つける作業は、変換プログラムを設計実装する際の負担となっている。この課題を解決するためには、パターン間でのイベントの出方の違いを迅速に発見し、その違いに相当するイベントのみに対して、作業との対応関係を定義する仕組みが必要である。

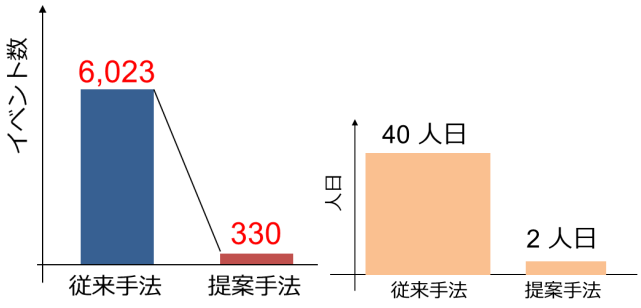
そこで、我々はイベント遷移モデルを用いて、パターン間におけるイベントの出方の異なる箇所を捉える手法を提案する。本報告では、この手法をイベント経路追跡アルゴリズムと呼ぶ。イベント経路追跡アルゴリズムは、あるパターンに対して設計者がイベント遷移モデルを作成した後、作成したモデルが別のパターンにおいてもイベントの流れを表すことができるかを判別する。

図 7 に、イベント経路追跡アルゴリズムを実施している最中の様子を示す。図 7 では、設計者がパターン 1 に対するモデルを設計した後に、イベント経路追跡アルゴリズムを実施している。イベント経路追跡アルゴリズムでは、パターン 1 とパターン 2 のイベントの出方の違いを捉えるため、パターン 2 のイベントの出方でも、イベント遷移モデルで、先頭のノードから遷移できるかを確認する。図 7 の場合は、ノード 5 において、イベント EVE18 で遷移ができないことが分かる。これは、パターン 1 内のイベントの出方に対し、パターン 2 のイベントの出方は、イベント EVE18 から異なることを意味する。そこでイベント経路追跡アルゴリズムでは、設計者に対し、ノード 5 から新たにイベント EVE18 に対応するノードおよびリンクを設置するように通知する。

設計者は、イベント経路追跡アルゴリズムで特定された、パターン間のイベントの出方の違いに対してのみ、ノードとリンクの追加、および作業を示す枠の設定を修正する。この後は、修正後のモデルが、パターン 2 のイベント EVE18 以降のイベントの出方を踏まえたモデルとなっているかを確認するために、イベント経路追跡アルゴリズムが、先ほど遷移が出来なくなったノード 5 から、イベント EVE18 以降のイベントで遷移し続ける。パターン 2 の最後のイベントまでを、設計したイベント遷移モデルで遷移し続けられたら、パターン 2 に関しては、設計が終了したものとする。イベント経路追跡アルゴリズムにより、従来はパターン 2 の全イベント (11 イベント) を設計者が確認して、作業の区切りを設定するところが、イベント EVE18 からイベント EVE17 までの 4 イベントの確認のみで区切りを設定できることになる。これは、パターン 1 で設定したイベントと作業の対応関係を、パターン 2 でも同じ個所があれば、その部分に対する設計を省略したことによる。これにより、設計者が作業の区切りを設定するのに要する時間が短縮される。

4. 評価

イベント遷移モデルを用いたモデル設計手法と、イベント経路追跡アルゴリズムにより、変換プログラムの設計において設計者がイベントログの中で確認したイベントの数が減少し、設計開発の時間が短縮されることを評価する。比較とする従来手法は、本提案手法を用いない方法であり、加工サイクル内のイベントの出方をパターン化し、パターン内の全イベントを設計者が確認して、図 3 に記した手順を実施するものである。



(a) 変換プログラムを作成するために確認が必要なイベントの数

(b) 作業の流れを捉えるために要した期間

図 8 本手法による変換プログラム作成の迅速化の評価

本評価ではまず、設計者が設計時に確認したイベントの数の減少を評価し、図 8(a) に結果を示している。従来手法と本提案手法のどちらも、同一のイベントログを用いて変換プログラムの設計開発を行っている。本評価では、金属旋盤加工工程における工作機械のイベントログを用いた。イベントログには、2 ヶ月分の稼動における、485,488 個のイベントが含まれている。本評価により、本技術を用いない場合、各パターンに含まれるイベントの総数である 6,023 個のイベントを確認する必要があった。それに対し、本技術を用いた場合、イベント経路追跡アルゴリズムにより、確認すべきイベントの数が 330 に減少し、約 1/18 に減少することを確認した。

また、確認が必要なイベントの数の減少による、変換プログラムの作成期間の短縮を評価し、その結果を図 8(b) に示す。変換プログラムの作成に要する期間は、本技術を用いない場合に 40 人日を要していたところ、本技術を用いることで、2 人日に短縮され、1/20 になっている。これにより、本技術は、生産現場への設備投資が不要な、低コストな作業監視の実現に貢献できるものと期待される。

5. 結論

近年 IoT 技術が注目され、製造分野においても、従来捉えることが困難であった生産現場の作業監視などに適用され始めている。しかし、多くの作業監視ソリューションは、システム構築コストが大きくなる問題がある。そこで、生産現場の設備が保守用に出力していたイベントログから、

作業を捉えるソリューションを検討した。本提案の手法により、イベントログから作業を捉えるための変換プログラムの作成に要する期間が、約 1/20 に短縮されることが確認された。これにより、イベントログから作業を捉え、分析する際の人件費も抑えられるため、本研究は、低コストで生産現場の作業監視を行うソリューションの実現に貢献できるものと期待される。

参考文献

[1] M. Swan, "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, pp. 217–253, Nov. 2012.

[2] L. G. Occhipinti, "Innovative manufacturing of large-area electronics," in *Proceedings of Solid State Device Research Conference (ESSDERC)*, pp. 194–197, Sept. 2014.

[3] S. R. Fletcher, T. L. Johnson, and J. Thrower, "A study to trial the use of inertial non-optical motion capture for ergonomic analysis of manufacturing work," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pp. 1–29, Aug. 2016.

[4] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Proceedings of Future of Software Engineering*, pp. 37–54, May 2007.

[5] J. R. Jovanovic, D. D. Milanovic, and R. D. Djukic, "Manufacturing cycle time analysis and scheduling to optimize its duration," *Journal of Mechanical Engineering*, vol. 60, pp. 512–524, May 2014.

[6] A. I. Sivakumar, "Optimization of a cycle time and utilization in semiconductor test manufacturing using simulation based, on-line, near-real-time scheduling system," in *Proceedings of conference on Winter simulation: Simulation—a bridge to the future*, pp. 727–735, Dec. 1999.