特集論文

阿部貴成*
Takaaki Abe
野口智史*
Satoshi Noguchi

製造設備向けローコード開発技術

Low－code Development Method for Manufacturing Facilities

*情報技術総合研究所

要　旨

製造設備の制御プログラムに対するローコード開発技術は，サイクル動作の設計手法として広く用いられているフローチャートから，サイクル動作を実現するための制御プログラムを生成する特徴を持つ。従来と設計手法を変えることなく生成を可能にすることで，設計工数を増やさずに制御プログラムの実装工数の削減が可能になる。また，設備開発を効率化するツールとして近年注目されている3Dシミュレーターとローコード開発技術を組み合わせることで，将来的に設備開発の更なる効率化の実現が期待できる。今後，このローコード開発技術のプロトタイプを社内の設備開発に適用して有効性を確認し，製品化に向けた開発を進めていく。

1．ま え が き

近年，労働人口の減少などの社会問題によって，工場の自動化に対する需要がますます高まっている。しかし，工場を自動化する設備の開発工数は大きいため，設備開発の効率化が求められている。設備開発の中でも，特に効率化が求められている作業の一つとして，設備を制御する制御プログラムの実装作業がある。制御プログラムの実装作業には，設備の設計を基に，似た処理を繰り返し記述することで実装する単純作業も多く含まれている。このような作業を効率化するために設備の設計から制御プログラムを生成する技術が求められている。

また，設備開発を効率化するツールとして近年，製造設備向けの3Dシミュレーターが注目されている。三菱電機も“MELSOFT Gemini”を販売している。3Dシミュレーターは設備をデジタル空間で表現することで，実機レスでの検証を可能にする。3Dシミュレーターを用いることで，従来は設備の組立て後に実施していた検証が設備組立て前に可能になり，設備開発のフロントローディングが可能になる。シミュレーションには先に述べた制御プログラムが必要である。しかし，現状の設備開発プロセスではそれを開発の早い段階で用意することが難しい。そのため，3Dシミュレーターを設備開発の最終段階でしか活用できないという問題がある。

そこで，本稿では，設備の設計から制御プログラムの生成を可能にするローコード開発技術を検討する。また，このローコード開発技術と3Dシミュレーターを組み合わせることで，設備開発の更なる効率化を実現する将来像を検討する。

2．制御プログラムのローコード開発技術

2.1　生成対象とする制御プログラムとその設計手法

製造設備は，ある一連の動きを繰り返し実行するサイクル動作を基本として自動生産を行っている。そのため，制御プログラムでは，設備に搭載されている各メカの動作順序や動作条件をコーディングして，サイクル動作を実装する必要がある。このサイクル動作のプログラムは，設備の自動生産の動きを定めているため，設備の生産能力に直接影響する重要なプログラムである。また，このプログラムは似た処理を繰り返し記述することで実装することが多いため，実装の効率化が求められているプログラムでもある。サイクル動作のプログラムは，これまで設計内容と実装内容の対応関係が整理されておらず，設計からプログラムを生成するような効率的な手法が提供されていなかった。そこでこの対応関係を整理することで，設計内容を反映したサイクル動作のプログラムを生成するローコード開発技術を検討した。この節では，サイクル動作のプログラムの実装手法として広く用いられている手法と，その設計手法について述べる。

国内での設備の制御プログラムは，PLC（Programmable Logic Controller）用標準規格IEC61131－3で規定された言語のうち，ラダー言語を用いて実装されることが多い。**図1**にその一例を示す（図右半分の“制御プログラム”の箇所）。ラ

ダー言語では，回路ブロックと呼ばれるリレー回路形式の処理を記述し，複数の回路ブロックを組み合わせて制御プログラムを実装する。また，ラダー言語では，サイクル動作の実装手法として状態遷移型記法と呼ばれる記法が知られている[1]。これは，状態遷移部と出力部の二つで構成される記法である。状態遷移部では，センサー信号などの値を基に設備のサイクル動作の状態を順次進める回路ブロックを記述する。**図1**中の状態遷移部で示すように，センサー信号(X1)のONなどの条件(**図1**中の黄色枠)が成立するごとに状態変数(M0～M2)を順次ONしていき，最後の状態変数がONすると全ての状態変数をOFFする回路ブロックを記述する。これによって，状態遷移後に初期状態に戻り，また初期状態から状態遷移を繰り返すサイクル動作のプログラムになる。

出力部では状態遷移部の状態変数の値を基に各メカと対応付く制御変数のON／OFFを切り替える回路ブロックを記述する。**図1**で示すように，制御変数Y10のON条件である状態変数M0がONしたらY10をONし，Y10のOFF条件である状態変数M1がONしたらY10をOFFするような回路ブロックを記述する。このような状態遷移部と出力部から成る状態遷移型記法を用いて設備のサイクル動作は実装されている。

また，このサイクル動作の仕様は，**図1**(“サイクル動作の設計”の箇所)に示すようなフローチャートで設計されるケースが多い。このフローチャートは，STARTから順次処理を実行しENDに到達したらSTARTに戻ることで，一連の流れが繰り返されることを前提としている。**図1**に示すフローチャートは，空気圧シリンダーで動作するプッシャーのサイクル動作を表現した例である。先に述べたようにサイクル動作のプログラムは，条件の成立によって順次状態を進めて，その状態によって制御変数のON／OFFを切り替える。そのため，処理の順序関係や条件を設計できるフローチャートを活用してサイクル動作を設計することが多い。

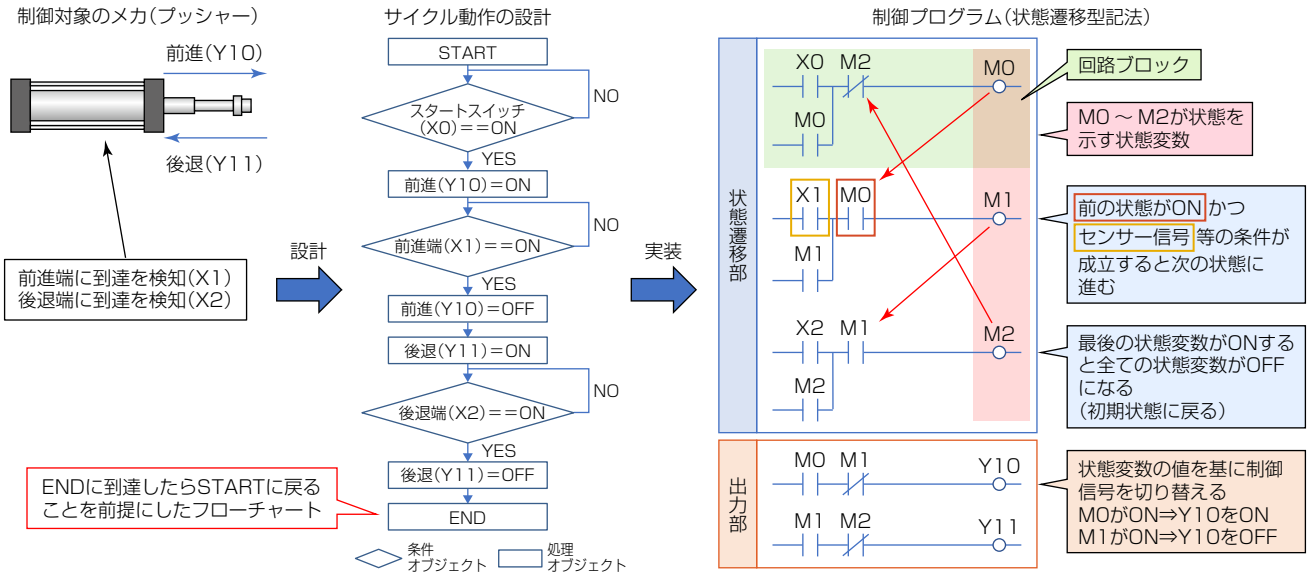

図1－サイクル動作の設計・実装例

2.2　ローコード開発技術の課題

制御プログラムのローコード開発を可能にするには，二つの課題がある。

第一に，**図1**で示した従来の設計手法であるフローチャートでサイクル動作を設計できることが必要である。ローコード開発技術が，これまでと大きく異なる設計手法を必要とする場合，設計工数が増大し効率化につながらない可能性がある。従来と近い設計手法であればこのローコード開発技術を活用した開発へシームレスに移行することもできる。

第二に，従来実装していた制御プログラムと近い形式の制御プログラムを生成できることも必要である。制御プログラムは実装者以外に保守員などもソースコードの閲覧や拡張をする。そのため，生成する制御プログラムの形式が変わると，これらの作業に支障が出て保守などの妨げになる可能性がある。

これらの課題を解決するために，従来の設計手法であるフローチャートから状態遷移型記法の制御プログラムを生成する技術を考案した。

2.3　制御プログラムの生成ステップ

ローコード開発技術では，フローチャートの設計内容と状態遷移型記法の回路ブロックをパターン化し，それらを対応付けることでフローチャートから状態遷移型記法の制御プログラムを生成する。この節では，この対応関係について述べた後，その対応関係を基に制御プログラムを生成する手法について述べる。なお，この節では説明を簡略化するために，フローチャートの設計内容のパターンと生成する回路ブロックのパターンを**図1**で示したものに限定することにする。実際の設備開発にこのローコード開発技術を適用する場合は，これらのパターンは複数ある。パターンの例としては，条件オブジェクトで進む先が分岐している場合や条件としてタイマーを用いる場合などが挙げられる。

まず，フローチャートの条件オブジェクトから状態遷移部の回路ブロックを生成する。**2.1節**で述べたように状態遷移部は，条件が成立すると次の状態へ移る特徴がある。そのため，条件が成立すると次のオブジェクトへ進むことを示す条件オブジェクトと状態遷移型記法の状態遷移部が対応付けられる。次に，フローチャートの処理オブジェクトから出力部の回路ブロックを生成する。出力部は状態遷移部の状態変数の値を基に制御変数のON／OFFを切り替える特徴がある。そのため，制御変数の値の書換えを示す処理オブジェクトの内容と状態遷移型記法の出力部が対応付けられる。制御変数の切替えは複数の処理オブジェクトに書かれるため，複数の処理オブジェクトと出力部の一つの回路ブロックが対応付けられる。

今回検討する手法では，これらの対応関係を基に次に示す三つのステップで生成する（**図2**）。

(1)　Step1：フローチャートの解析

フローチャートを解析し，フローチャートの各オブジェクトに対して，オブジェクトの種類やオブジェクト間の結線の情報から生成すべき回路ブロックのパターンを特定する。**図2**に示すようにオブジェクトの種類を基に，条件オブジェクトに対して状態遷移部の回路ブロック，処理オブジェクトに対しては出力部の回路ブロックのパターンであると特定する。状態遷移部の先頭の回路ブロックは全状態をリセットする処理を含み，先頭以外の回路ブロックと異なる。そのため，条件オブジェクトのうちフローチャートの先頭にあるものを結線の情報から特定し，**図2**左側に示すように“状態遷移部(先頭)”のパターンであると特定する。

(2)　Step2：変数の割り振り

状態遷移部の一つの処理に対して一つの状態変数が必要であるため，各条件オブジェクトに対して状態変数を一つ割り振る。**図2**の条件オブジェクトに対してM0から順に状態変数を割り振る。

(3)　Step3：各オブジェクトと対応する回路ブロックを生成

Step1で特定したパターンに応じた回路ブロックの雛形(ひながた)に対して，フローチャート上の情報や割り振った状態変数の情報を基に，フローチャートの内容を反映した回路ブロックを生成する。条件オブジェクトに対しては，条件オブジェクトに記載されている条件や，オブジェクト間の結線の情報などから雛形に設定すべき変数を特定し，その変数を設定することで回路ブロックを生成する。処理オブジェクトに対しては，制御変数ごとに複数の処理オブジェクトについての結線の情報から必要な変数を特定し，その変数を設定することで回路ブロックを生成する。

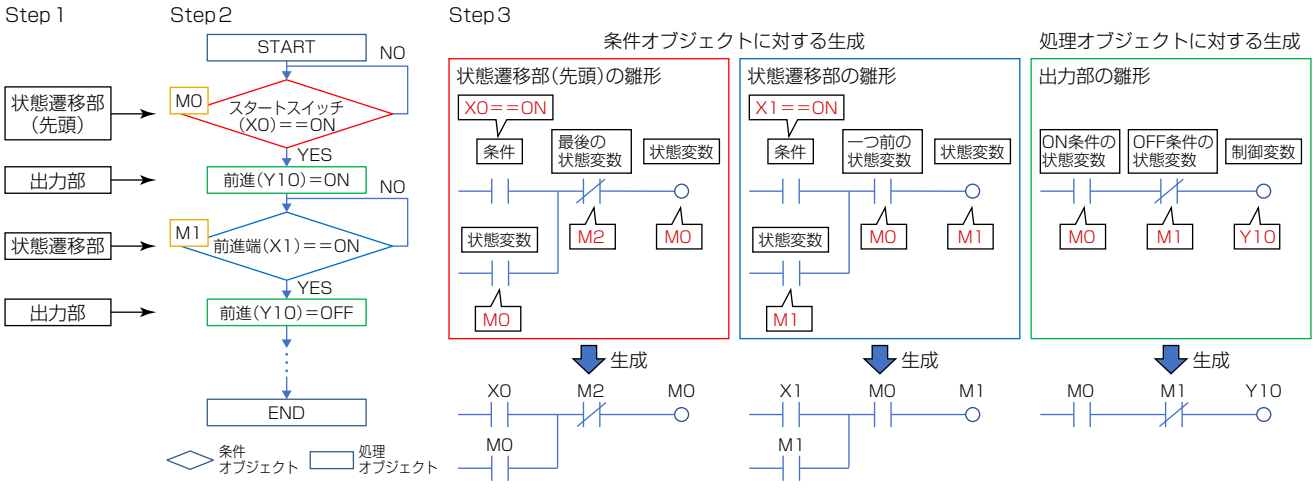

図2－制御プログラム生成の処理イメージ

このようにして，フローチャートの設計内容と対応する回路ブロックを生成し，生成した回路ブロックをフローチャートと対応付くように並べることで状態遷移型記法の制御プログラムを生成する。

3. 製造設備向け3Dシミュレーターを活用した設備開発

3.1 製造設備向け3Dシミュレーター

3Dシミュレーターは，設備をデジタル空間で表現することで，実機レスでの検証を可能にするツールである。3Dシミュレーターには，その用途の一つに装置シミュレーション[(2)]がある。装置シミュレーションとは，設備の形状(3D-CAD)や設備に搭載されているプッシャーやコンベヤー等のメカの動きを3Dシミュレーター上で定義し，設備の動きをデジタル空間で再現するシミュレーションである。

装置シミュレーションを実施するには，まず，3D-CADのデータなどを3Dシミュレーターにインポートして設備の形状を設定し，続いて，設備に搭載されている各メカの動作方向や動作速度などを設定する。そして，3Dシミュレーター上のメカを制御するために，制御プログラムのシミュレーターと3Dシミュレーターを連携させて装置シミュレーションを実行する。

3.2 3Dシミュレーターを活用した設備開発の課題

3Dシミュレーターの運用上の課題は，制御プログラムの用意に時間がかかるために開発の早い段階で3Dシミュレーターを活用できていないことである。この課題には二つの要因がある。一つ目は，現在の設備開発では，設備の仕様や機械図面を設計する機械設計と，制御プログラムを設計する制御設計で開発組織が分かれていることが多いためである。また，二つ目は，開発の流れとして機械設計での設計内容を基に制御設計をすることが多いためである。これらの要因から，プログラミング経験の少ない機械設計者は制御プログラムの実装が困難である。また，制御プログラムの実装が設備開発の最終段階であるために，3Dシミュレーターを開発の最終段階でしか活用できない問題が起きている。

3.3 ローコード開発技術と3Dシミュレーターを用いた将来の設備開発

本稿で検討するローコード開発技術と3Dシミュレーターを組み合わせた設備開発の流れを**図3**に示す。ローコード開発技術を活用することで，制御プログラムの実装に関する知識が少ない機械設計者でも効率良く制御プログラムを実装できるようになる。機械設計の段階から3Dシミュレーターの活用が可能になれば，精度の高い設計を効率的に行えるようになる。

例えば，従来の機械設計では，設備の動作の一例を表現したタイミングチャートを作成して，タクトタイムの見積りや設計者間の認識共有を行っていた。それに対して，このローコード開発技術を活用して3Dシミュレーターを機械設計の段階から活用できると，精度の高いタクトタイムの検証が可能になり，また，設計者間の認識合わせも効率化できる(**図3**左側)。

制御設計でもローコード開発技術によって効率良く制御プログラムが実装できる。3Dシミュレーターによって実機レスで検証ができるため，サイクル動作の設計のブラッシュアップを制御設計者が効率良く実施可能になる(**図3**中央)。

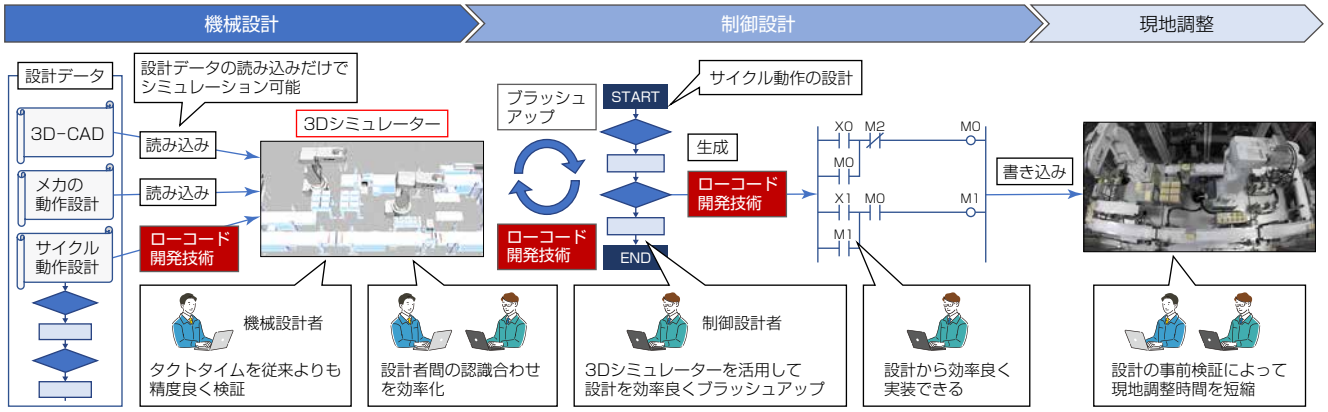

図3-ローコード開発技術と3Dシミュレーターを用いた将来の設備開発

4. む　す　び

フローチャートからサイクル動作の制御プログラムを生成するローコード開発技術を述べた。このローコード開発技術によってプログラム作成工数の削減ができるだけでなく，3Dシミュレーターを設備開発の早期に活用でき，設備開発全体の更なる効率化が可能になる。今後は，このローコード開発技術のプロトタイプを社内の設備開発に適用して有効性を確認し，この技術の製品化に向けた開発を進めていく。

参 考 文 献

(1) 必携PLCを使ったシーケンスプログラム定石集装置を動かすラダー図作成のテクニック，熊谷英樹，日刊工業新聞社（2021）
(2) 三菱電機：生産ライン・装置シミュレータならMELSOFT Gemini
https://www.mitsubishielectric.co.jp/fa/topics/2022/09_3dsim/index.html