

プラント操業支援システムとワークフローの抽出手法

筑波大学大学院 倉橋節也, 東京電機産業株式会社 飯島勉

1. はじめに

熟練者の大量退職という問題は、製造現場に品質管理の変革をもたらそうとしている。高機能化学材料など多品種少量で高品質な化学製品の製造には、高度な自動制御システムに加えて、熟練オペレータによるプロセス状態の確認と必要に応じた手動操作が欠かせない。しかし団塊の世代と呼ばれる大量の熟練者が定年を迎え、長年の経験による優れた技能を持つ彼らが、製造現場から去りつつある。一方、国際競争の中、製造現場で働く作業者の多くは需要に柔軟に対処できる派遣スタッフとなり、正規スタッフは半分近くに減少してきている。現在、多くの製造現場で、どのように熟練者の技能を伝承していくのかが大きな課題となっている。

従来は、個々の作業を膨大な標準作業手順に記述し、作業者はそれに従ってプラント操業や巡回検査を行うことが求められてきた。しかし、投入原料の成分変動や運転条件などによって、プロセスの状態は大きく変化してしまう。せっかくの標準作業手順書も、非定常時や異常時の例外的な処置まではすべて記述されない。本稿では、非定常時を含む運転支援に必要なさまざまなパラメータを一括管理するシステムと、運転に必要なワークフローを実プロセスデータから抽出し運転支援システムに実装することによって、これらの問題に取り組んだ事例を紹介する。

以下 2 章においてバッチ製造工程における課題を述べ、3 章で銘柄管理と運転支援システムの連携について、4 章でシステム導入事例を紹介し、5 章で運転支援システム構築の課題、6 章で熟練者からワークフローの抽出手法、7 章でまとめと今後の課題について述べる。

2. バッチ製造工程における課題

自動化が進んだ今日でも、人の判断に頼った操業は未だに多く見受けられる。これらの原因は、においや粘性の「べとべと」「ねとねと」といったセンサーでは検知できない熟練者の微妙な判断や勘や頼っているためである。また、いずれのシステムでも製造記録として日報やバッチ報といった帳票を作成しているが、これらの感覚的な情報は帳票内に手書きで記載していることが多く、熟練者のノウハウがシステムに取り込まれることがほとんど無いのが現状である。対策として作業手順書によるノウハウの伝承を行っている事例は多々あるが、上記のように、作業手順書による作業指示は、あらゆる状態に対応できるよう記載されていることは稀であり、オペレータの適応に依存せざるを得ない。また作業指示と実績記録が連動したシステムもコスト面の問題があり、そう多くない。

3. 銘柄管理と運転支援システムの連携：

これらの課題に対して、横河電機製の運転効率向上支援パッケージ「Exapilot」と東京電機産業製の簡易銘柄管理パッケージ「CS-Recipe」、バッチ帳票パッケージ「BatchRepo」を紹介する（図1）。

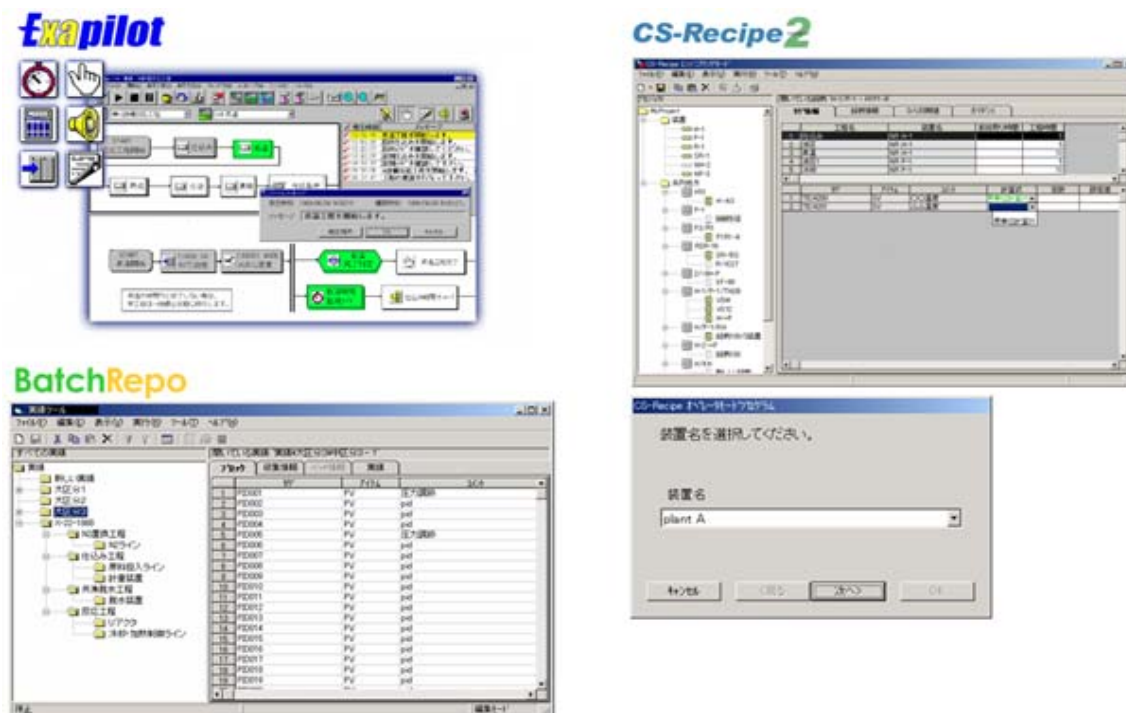


図1 Exapilot, CS-Recipe, BatchRepo の画面表示例

Exapilot はフローチャート上に作業手続きを登録し、非定常化業務・定常化業務・バッチ運転を支援するパッケージソフトウェアである。フローチャートに記述することで直感的な手順定義が可能である。言語プログラムの知識も不要でありながら、高度な業務支援が可能となる。Exapilot と DCS は OPC インターフェースで接続され、Exapilot 側から自由に DCS のデータにアクセスすることが可能であり、DCS に対して運転を支援することもできる。また、OPC サーバーの機能を有しており、外部アプリケーションとの接続が容易である。更に Visual Basic Script (以下 VBS)アイコンを使用することで、プログラムの起動やコントロール、データの交換といった複雑な接続も行うことができる。

CS-Recipe は横河電機の DCS CENTUM CS3000 用の銘柄管理パッケージとして開発しており、エンジニアリングモード画面とオペレータモード画面の2画面から成り立っている。エンジニアリングモード画面では Windows の Explorer 風な操作性を持ち、銘柄管理に必要な全ての定義を行うことができる。銘柄データのダウンロードバッファの集まりである「装置」という単位で登録し、各装置内にレシピデータを持つ「銘柄」を作成する。一方、オペレータモード画面はオペレータ用のレシピデータ設定画面である。表示内容はオペレータが操作に必要な情報に絞り、ウィザード形式で選択していくことで、簡単に銘柄設定を

行うことができる。レシピ設定は OPC にて行われる。

BatchRepo も CENTUM CS3000 に対応したバッチ帳票専用のパッケージである。定義状態（エンジニアリングモード）と収集状態（収集モード）の 2 つのモードを持っている。エンジニアリングモードでは CS-Recipe のエンジニアリング画面と類似した画面で帳票収集タグの定義を行い、実際の帳票のレイアウト作成は Microsoft Excel のテンプレート機能を使用し作成する。定義後は収集モードに切替え、バッチ開始を待機状態となる。バッチ進捗に併せて OPC でデータを収集、バッチ終了時に自動で帳票印字を行うといったことが可能である。すべてのパッケージに共通して、言語によるプログラミングの必要はない。

4. システム導入事例

次に Exapilot と CS-Recipe/BatchRepo とを連携させた事例を紹介する（図 2）。

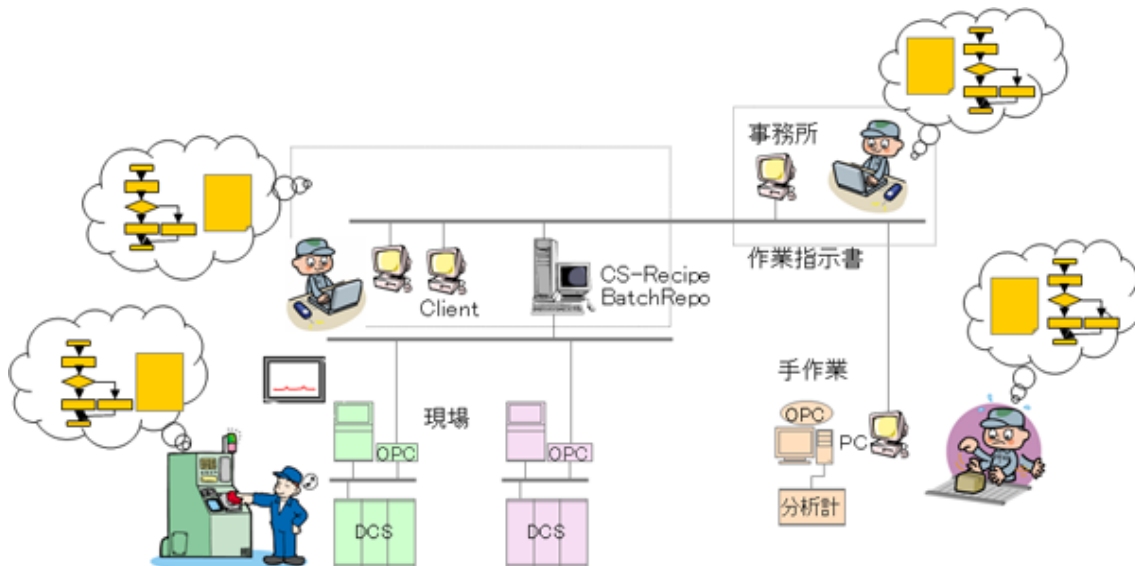


図 2 Exapilot と CS-Recipe/BatchRepo のシステム連携例

CS-Recipe や BatchRepo は DCS との接続を前提に開発されているが、接続のインターフェースとして OPC 以外にも VB/VBS/コマンドシェルによる指示、起動、銘柄データの公開などといった多様な接続性を持っており、Exapilot の VBS アイコンを使うことでより効率的に接続することができる。事例の動作概要は以下のとおりである。（図 3 参照）

- 1) Exapilot の業務と呼ばれるプログラム（以後、業務）から CS-Recipe の銘柄一覧を取得
- 2) 業務から CENTUM CS3000 へ取得した銘柄一覧を書き込む
- 3) CENTUM CS3000 のグラフィックを介して、銘柄の選択を行う。
- 4) 選択した銘柄のレシピデータを業務の指示で、CS-Recipe から CENTUM CS3000 へダウンロードする。
- 5) バッチ開始とともに、業務が BatchRepo に対して帳票収集の要求を行う。

- 6) 以後、工程進行のたびに **CS-Recipe** に対しては次工程のレシピデータの要求を、**BatchRepo** に対しては帳票データの収集要求を行う。
- 7) 帳票に手書きで記録していたオペレータの判断を **Exapilot** で入力し **BatchRepo** に書き込む。
- 8) バッチ終了により、印字起動を **Exapilot** から行う。
- (1,4,5,6,8 については VBS の指示で、2,3,4,7 については OPC 接続で行った。)

今回の事例では **Exapilot** から、**CS-Recipe** の銘柄情報を活用した事例であり、本来の操作画面であるオペレータモード画面は使用していない。また、すべてのレシピの操作を **Exapilot** で行うことで、複数のパッケージの組み合わせにより操作が煩雑になることを防止する狙いもある。このシステムでは **BatchRepo** は単に帳票を印字するだけでなく、取り込んだバッチデータのすべての情報をデータベースに記録しているため、**DCS** に取り込まれていない熟練者の判断情報もバッチデータと共に保存することが可能になった。

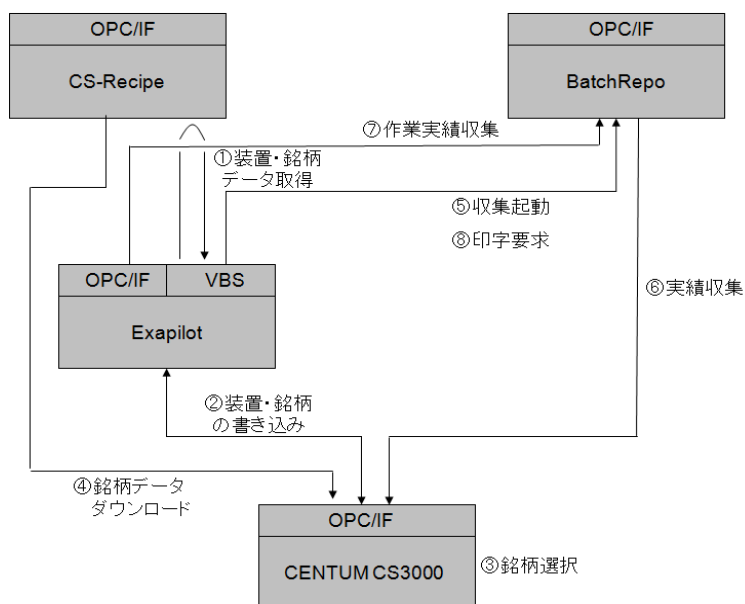


図3 連携システムの動作フロー

5. 運転支援システム構築の課題

以上のような運転支援システムを、実際に現場へ導入する場合に重要なことは、知識の抽出にある。標準作業手順書には、非定常時や異常時などの重要な熟練者の操作手順が表現されていない場合があり、実際に運転支援システムを導入した企業では、このような運転ワークフローを抽出するために、多大な労力をかけていることが報告されている。筆者等は、これらの知識抽出のための手法を開発した。以下で、その手法のプラントへの適用事例を報告する。

6. 熟練者からワークフローの抽出手法

本手法は、複雑なプラント操業時系列データから、熟練者の操作情報を取り出すことを目的とする。プロセスデータ解析のためのモデル構築のステップを以下に示す。

1) プロセスデータ収集

プロセスデータを定周期で収集し、プロセスデータベースに時系列データとして格納する。

2) データの正規化

収集したプロセスデータのスケールはそれぞれ異なるため、0.0~1.0に正規化する。

3) 相互相関分析

正規化された二つのプロセスデータを選択し、最大の相関を示す時間差を探索する。

4) プロセス応答モデル

シフト時間と相関係数から、プロセスデータ間の関係を応答モデルとして記述する。

5) 制御ルールの発見

プロセス応答モデルから、ルール学習システムを実行し、制御ルールを見出す。

6) ワークフローの発見

運転操作イベントとプロセスデータに対して、学習システムを実行しワークフローを見出す。

連続プロセスデータの場合、時系列データ間に広く相関が認められる。そこで、データ間の関係に着目した相互相関係数を求めることで、図4のようなプロセス応答モデルを得ることができる。

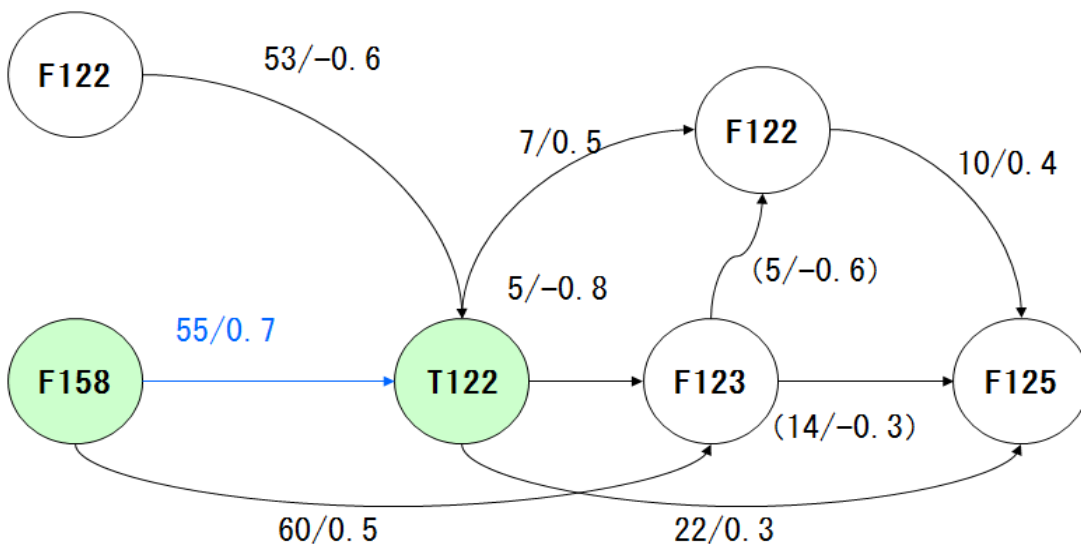


図4 プロセス応答モデル

また、オペレータが状態を早急に把握し、制御を行うために、簡潔で明快なルールの発見が必要となる。そこで、IF~THENルールを効率よく発見できる学習システムを開発した。このシステムは、ランダムに生成したルールでプロセスデータを分類し、その結果を評価

することで新たなルールを生成する方法である。数値データは、PV、下降傾向、安定、上昇傾向に分類してルール発見を行う。プロセス応答モデルで得られた相関の高いタグのデータを対象に学習システムによる制御ルールの探索を行った結果、以下のようなルールが発見することができた。

(75% < F2) and (75% < F3) then 50% < T2

プラントの熟練者運転員は自動制御操作以外に、自らの判断で設定値や操作量を随時変更している。そこで、これらの熟練者の操作を発見するために、操作ワークフローを生成する手法を開発した。はじめに、バルブの開閉などのイベントデータと、プロセス制御のための圧力などの設定値を操作したタイムスタンプ付データを収集する。次に、非定常時のデータから、上記の手法を用いて制御ルールを探索する。発見されたこれらの制御ルールは、タイムスタンプが付いていることから、操作時間順に並べ替えることができる。図5は、学習システムを実行した結果である。

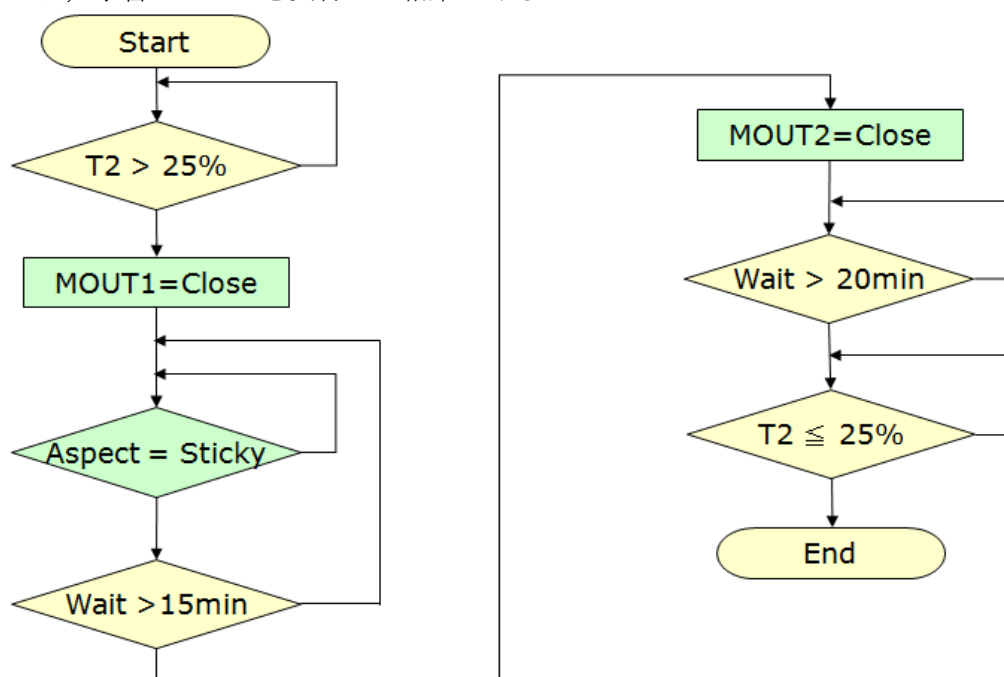


図5 学習システムにより生成された操作ワークフロー

このワークフローは小規模ではあるが、手動操作記録とプロセスデータからオペレータの操作を抽出したものとなっている。これらの操作は、普段は意識されていないものであり、熟練運転員の暗黙知を示唆するものと考えることができる。本手法の採用により、ベテランの運転員から未熟練運転員へ熟練技能を伝承していくためのヒントを提示することが可能となる。

7. まとめと今後の課題

本稿では、OPC 技術を用いたオープンなバッチプラント向けの銘柄管理システムと運転

支援システムの連携について紹介し、それらを実プラントに導入するにあたっての、プロセスデータから運転操作ワークフローを生成する手法を紹介した。そして、実際のプラントのデータから、熟練者の運転技能を抽出できることを示した。この手法は比較的単純な方法ではあるが、プロセス時系列データから暗黙知としての熟練者の運転操作を抽出し、運転支援システムに定義することができ、非熟練者への技能伝承に役立つものとする。提案した手法は、更に多くの実データを用いて実証を行い、より使いやすい製品に向けて開発を行う予定である。

参考資料

日本鉄鋼協会 計測・制御・システム工学部会フォーラム「技術・技能をいかに伝えるかー事例報告と研究動向ーその2」, 2009