

《特集：プラントエンジニアリングの情報化と支援ツール》

OmegaLandとExapilotによる 運転支援シミュレーション

(株)オメガシミュレーション 横山 克己
Katsumi Yokoyama
横河電機(株) 佐藤 恵二
Keiji Sato

1. はじめに

最近では製品の提供がソリューション志向になり、顧客の要求に対して他社のハードウェアやソフトウェアを組み合わせる機会が増えている。パソコンやOSの仕様が統一されたと同時に、プラントデータの活用にOPC (OLE for Process Control) のようなインタフェースの標準化も進んで、比較的容易にしかも安価に構成できるようになってきている。本稿ではそのようなコラボレーションの1つの例として、統合ダイナミックシミュレーション環境OmegaLandと運転効率向上支援パッケージExapilotを組み合わせた、運転支援シミュレーションを紹介する。

OmegaLandは本誌ですでに紹介済み⁽¹⁾⁽²⁾であるが、ダイナミックシミュレーションを用いてプラント運転のいろいろな場面に応用できる統合環境で、(株)オメガシミュレーションが提供している。一方、Exapilotはプラントのスタートアップ、シャットダウン、ロード変更や銘柄変更など非定常の運転操作を自動化できるパッケージであり、横河電機(株)が開発、販売している⁽³⁾。

Exapilotはオンラインで使われるパッケージであるため、実プラントに適用してその動きを確認する必要があるが実プラントの確認範囲は限られてしまう。そこで、OmegaLandのダイナミックシミュレータで仮想プラントを構築し、実プラントに適用する前にExapilotで作成した運転手順の検証を行うことができる。

以下では、まずそれぞれの製品を簡単に紹介したのち、本題である運転支援シミュレーションについて事例を交えて説明する。

2. OmegaLandの機能と特長

OmegaLandはダイナミックシミュレータ Visual Modelerをコアモジュールとし、それを用いた応用システムを構築するために必要となる機能、たとえばデ

ータベース機能やグラフィック機能、通信機能などを機能単位ごとにモジュール化して整備した統合環境である。それらのモジュールを組み合わせることによって、様々なシミュレータを構築することができる。その特長は以下の4つある。

- ① パソコンで動作：OmegaLandはWindows 2000/NTに対応しており、パソコンで使うことができる。
- ② 多目的に利用可能なモジュール構造：求められる機能を独立性の高いモジュールにして開発しており、ユーザの多様なニーズに応えられる組み合わせが可能となっている。
- ③ 高精度のシミュレーションを実現：世界的に見ても最高レベルのダイナミックシミュレータ Visual Modelerをコア技術として使用している。
- ④ オープンなインタフェース：ユーザがモジュールを開発できるように、モジュール間を接続するインタフェースが公開されており、開発キットが提供される。さらに業界標準のインタフェースであるOPCに対応している。

OmegaLandは、ダイナミックシミュレーション技術をプラント運転に活用する環境を提供するものであるが、機能モジュールの組み合わせで以下のような場面への応用が可能である。

(1) ビジュアライゼーション

Visual Modelerで実行したシミュレーション結果を、グラフィカルでわかりやすく表現できる。

(2) プロセス設計・解析シミュレーション

起動・停止などの運転方法を変更した場合の安全性の確認、排出物質の挙動シミュレーションによる環境への影響の評価、PIDチューニングの解析、プラントの原単位計算や、プラントのライン/装置/制御機器などの改造を行った結果の評価に利用できる。

(3) 教育シミュレーション

装置内の状態や操作量や制御変数を変更した場合の挙動などをグラフィックやサウンドで表現することにより、プロセスおよび制御システムの原理や運転変数への理解を深めることができる。

(4) 運転訓練シミュレーション

実プラントと同じ臨場感溢れる運転操作環境を利用して、プラントのスタートアップ、シャットダウン、異常時対応、定常時運転における種々のプラント操作に対応できる運転員育成のために、即戦的な運転訓練が行える。

(5) 制御支援シミュレーション

外部ソフトウェアとの接続を可能とすることにより、それらのソフトウェアでの制御モデル開発や制御性の検証の負荷を軽減するとともに設計結果の信頼性を向上させることができる。

(6) 最適化シミュレーション

予測シミュレーションによる運転支援や連続プラントの非定常操作の最適化、および自動実行と運転評価機能により、ケーススタディを通じて最適な運転操作条件を見つけ出すなどのプラント運転をより最適化するための環境を提供する。

本稿ではこのなかで、(5)の制御支援シミュレーションを取り上げるが、このシミュレーションを行う環境がVMassistant（制御支援シミュレーション環境）である。VMassistantは、プロセスモデルモジュール（Visual Modeler）、実行制御モジュール（EXEC）、データベースモジュール（DB）、グラフィックモジュール（VIEW）の4つのBasic機能モジュールに、インタフェース機能をもつPCS接続モジュール（PCS）とOPCインタフェースモジュール（OPC）の2つのオプション機能モジュールを加えて、計6つのモジュールで構成される。

3. Exapilotの機能と特長

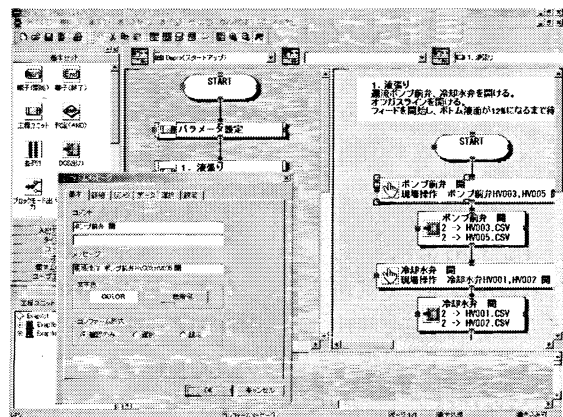
Exapilotは、銘柄変更やスタートアップ／シャットダウンといった手動操作主体の非定常運転を高いレベルで標準化することを目的に開発されたパッケージである。Exapilot上で熟練運転員が自らの運転ノウハウを自らの手で標準化することにより、運転員のスキル差による原材料／時間／ユーティリティ使用量等のロスを削減し、運転効率の向上を実現することができる。

Exapilotの最大の特長は、運転員自身によるアプリケーションの開発を可能にするほどの分かり易さ、操作性の良さであり、これらは以下に示すような機能に

より実現されている。

(1) 構築機能

- 業務（Exapilotのアプリケーションの単位）は、第1図のような直感的なフローチャート形式で記述される。



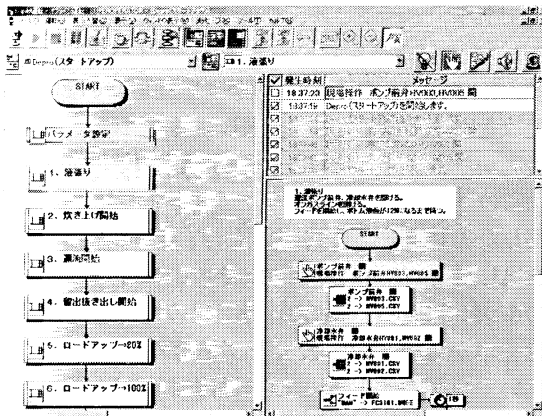
第1図 Exapilot構築画面

- ポンプスタート、MVランピング、変化速度確認、メッセージ出力等の運転操作に必要な機能があらかじめ工程部品／モジュールとして提供されている。
- 自動運転（DCS/PLCより信号が取り込まれているもの）と手動運転（操作を促すメッセージを出力するもの）を同じ業務フローに記述することが可能である。
- ユーザが作成したフローチャートをモジュール化し、再利用することが可能である（ユーザ工程モジュール）。

- Windows上で動作するパッケージのため、他のパッケージ（MS Office等）との連携が容易である。

(2) 運転機能

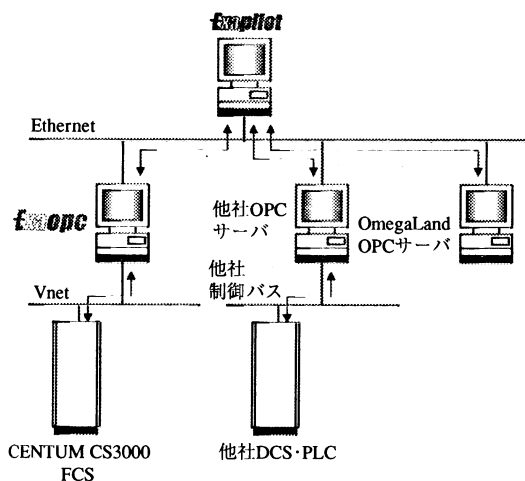
- 第2図のように構築機能にて作成した業務フローがそのまま、運転用の画面として利用される。
- 工程部品の状態（実行中／終了等々）を色で表現することにより、運転状態を直感的に把握することが可能である。
- オフライン、試運転、通常運転の3つの運転モードを利用することで、DCS/PLCが無くてもデバックから実用運転にスムーズに移行することが可能である。
- 工程開始、工程終了、工程一時停止、工程ブレイク、工程スキップの運転コマンドが用意されているため、運転状況に柔軟に対応した運転が可能



第2図 Exapilot運転画面

である。

- 弊社のDCSに限らず、OPCインタフェースをサポートした制御機器であれば接続することが可能である（第3図）。



第3図 Exapilot構成図

4. 運転支援シミュレーションの構成

Exapilotに代表される運転支援パッケージや高度制御向けの多変数モデル予測制御パッケージなどを現場に適用するには、事前に十分なテストならびにチューニングが必要であるが、VMAssistantを利用することにより現場の運転に影響を与えることなく繰り返し検討を行うことができる。また、これらのシステムで作成した制御ロジックや運転操作がプロセスに与える影響をシミュレーションによりダイナミックに確認できる

ため、プロセス改造に伴う制御ロジックの変更や運転効率化のための複雑な運転手順を作成する場合には、その検討手段としても有効である。

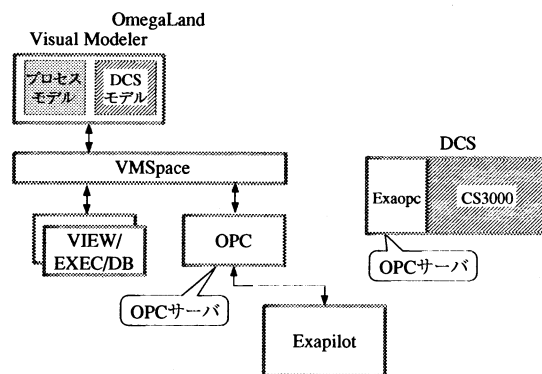
ExapilotなどこれらのパッケージとOmegaLandとは、OPCインタフェースによって容易に接続することができ、OPCサーバあるいはOPCクライアントのいずれの接続方法も可能となっている。

運転支援シミュレータは、DCS（分散制御システム）をシミュレータでエミュレーション（模擬）するか、実機DCSを接続するかで2通りの実現方法がある。

なお、以下の説明ではDCSには横河電機(株)製DCSであるCENTUM CS3000を例示した。

(1) DCSエミュレーションタイプ

第4図にその構成図を示した。左側のOmegaLandではVMSpaceというデータ・制御バスの仕組みを利用して、先に挙げた機能モジュールが連携して動作する。ExapilotとはOPCインタフェースモジュール（OPC）で接続されており、このモジュールがOPCサーバに、そしてExapilotがOPCクライアントの関係になっている。そして、左側に示したDCSに相当する部分はVisual Modeler上に制御モデルとして構築されている（対応する部分としてハッチングして示した）。

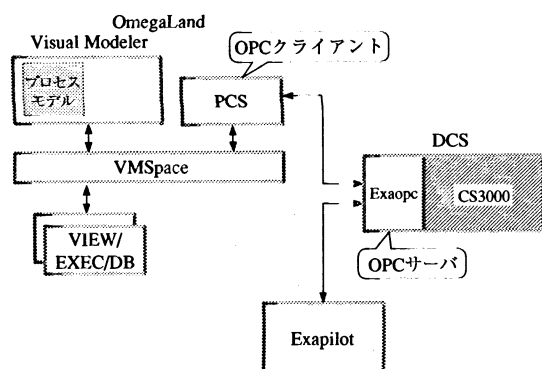


第4図 DCSエミュレーションタイプ

このタイプでは制御モデルの構築が必要ではあるが、DCSを必要としないため、導入が容易で手軽に検討することができる。Exapilotのエンジニアリングが終わって実プラントに適用するときには、Exapilotの接続をOmegaLandからDCSに切り替えて、DCS側のOPCサーバにつないで運用することになる。

(2) 実機DCS接続タイプ

第5図の構成図がこのタイプである。第4図と比較するとわかるように、Visual Modelerには制御モデル



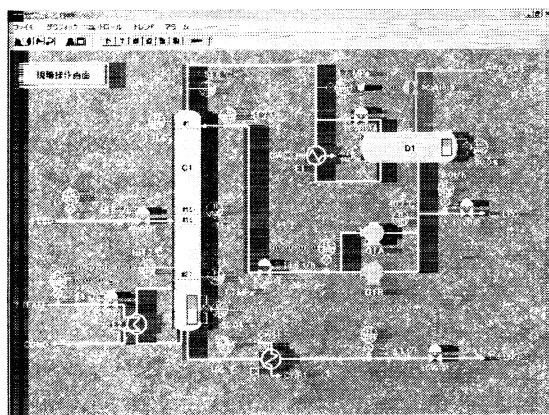
第5図 実機DCS接続タイプ

はなく、プラントモデルを実機DCSが直接制御することになる。また、DCS側のOPCサーバに対して、OPCクライアントとしてPCS接続モジュール（PCS）を接続し、実行周期ごとにデータ交換を行う。DCSには実プラントと同じものが使えるため、調節計のパラメータ設定値をそのまま使えるうえ、Exapilotは通常の運用と同じく、DCSの値を監視したり、設定値をセットしたりできるので、シミュレータを仮想プラントとして使う本来の使い方といえる。ただし、シミュレーション環境用のDCSを実プラントとは別に用意しなければならない。

5. 適用事例

ここではDCSエミュレーションタイプでExapilotとOmagaLandを接続した事例を紹介する。

対象としたプラントモデルは石油化学で使われる分離装置のモデルで、C3（プロパン）とC4（ブタン）を分離する蒸留塔である。第6図にこのプロセスのVIEW画面（運転監視画面）を示した。原料は蒸留塔の中段



第6図 脱プロパン塔（デブタンナイザー）

から塔内へフィードされ、塔底に設置されたりボイラにより炊き上げられて、気液平衡操作によって分離される。塔頂からは軽質留分C3がコンデンサで冷却されて抜き出され、一部は還流として蒸留塔に戻される。塔底からは重質留分であるC4が抜き出される。

このプロセスのスタートアップ業務をExapilotにて構築した。工程を液張り、炊き上げ、還流開始、留出抜き出し開始、ロードアップ80%そしてロードアップ100%の6つに分けて作成した。各工程の操作にはDCSの操作卓から行う操作と、ポンプ起動のように現場に向いて行う操作がある。DCS操作はオペレータが行う操作をExapilotにそのまま記述して構築した。先に示した第1図が液張り上げ工程を編集集中のExapilotの画面であり、第2図が同じく実行画面である。

仮想プラントを相手にしているため、試行錯誤が可能で何回もやり直しがきくうえに、加速運転の機能を使って短時間で確認作業ができる。このスタートアップ運転は通常2、3時間を要するが、仮想プラントでは30分程度で実行できた。このようにして、経験できる機会が少なくなったスタートアップ運転について、事前にその操作手順を整理し、より少ない時間で確実にスタートアップできる手順の確立ができることを確認した。

6. おわりに

OmagaLandとExapilotを連携した事例を説明してきた。パソコンのシステムでは、他の特徴的な機能をもつシステムやパッケージと有機的な結合をして機能を高度化したり、お互いにデータをやり取りして利用する利便性が求められる。OmagaLandおよびExapilotははじめからオープンなインタフェースを考慮して作られており、様々なインテグレーションが可能となっている。今後も新規なシステムやツールとの組み合わせを提案していきたい。

※OmagaLandは(株)オメガシミュレーションの登録商標である。

※Visual Modelerは三井化学(株)の登録商標である。

※CENTUM、Exapilotは横河電機(株)の登録商標である。

※Windows、MSなど、本文中で使われている会社名、商品名は各社の登録商標あるいは商標である。

<参考文献>

- (1) 石川真紀夫, OmegaLand<統合ダイナミックシミュレーション環境>, 配管技術, Vol.42, No.5, P.11 (2000)
- (2) 横山克己, 統合ダイナミックシミュレーション環境OmagaLandとその応用, 配管技術, Vol.43, No.6, P.30 (2001)
- (3) 川村春久, 運転支援から操業支援へーその考え方と提案, 計装, Vol.44, No.2, P.23 (2001)

(筆者紹介はkeyman flash 参照)