

# OmegaLandのコア技術

## - ダイナミックシミュレータVisual Modeler -

### Dynamic Modeling Module of OmegaLand "Visual Modeler"

湯本 隆雅<sup>\*1</sup> 小口 梧郎<sup>\*1</sup>  
YUMOTO Takamasa OGUCHI Goro

Visual Modelerは、株式会社オメガシミュレーションが開発した新しいコンセプトによるプラントの統合ダイナミックシミュレーション環境OmegaLandの中核であり、実時間でプラント全体のシミュレーションを可能とする「プラントモデルモジュール」である。バルブ、熱交換器、ドラム、蒸留塔など各種プロセスユニットと流量計、温度計、PID制御装置など各種計装ユニットをPFD(プロセスフロー図)イメージで配置し、物質の流れ(ストリーム)と情報の流れ(信号線)とでユニット間を接続して、プロセスモデルを構築する。Visual Modelerは、シミュレータ構成の基本技術としてモジュラーアプローチを採用すると共に、プロセスのユニットモジュール間の圧力と流量の調整を、厳密且つ高速に行う計算法(圧流バランス計算法)を開発することによって、これらの要求を満足するシミュレーションを実現している。

Visual Modeler is the "Plant Model Module" that Omega Simulation Co., Ltd. has developed as an integrated dynamic simulation environment of a plant, and the core of "OmegaLand" that enables simulation of the entire plant at realtime. It constructs a process model in the way that various process units including valves, heat exchangers, drums, and distillation towers, and such instrumental units as flowmeters, thermometers, and PID control devices are allocated in PFD (Process Flow Diagram) images and connected with material flows (streams) and information flows (signal lines). The Visual Modeler has realized high-performance simulation which satisfies the requirements of OmegaLand, with a modular approach as basic technology for simulator configuration and the strict and speedy calculation method (the method of computation for pressure and flow balance) for adjusting pressure and flow among unit modules of the process.

#### 1. はじめに

OmegaLand<sup>2</sup>では、多様なプロセスと各種の用途に適用可能な柔軟性と精密性、リアルタイムシステムとして利用可能な高速性とロバスト性を兼ね備えたシミュレータが必要である。その機能を提供しているのがVisual Modeler<sup>3</sup>である。本稿では、Visual Modelerにおいて実時間ダイナミックシミュレーションを可能とする仕組みや圧流バランス計算法を中心に紹介する。

#### 2. Visual Modelerによるプロセスモデルの編集

Visual Modelerでは、独立性を持ったプロセスモデルを複数組み合わせ、大規模なプラントモデルを編集可能にしている。また、プロセス単位で編集と実行をサポートしているため、柔軟な設計、開発、実行が可能である。

始めに、プロセスモデル毎に利用する物質成分のリス

トを作成する。次に、ユニットやストリームで利用する部分成分リストとエンタルピ、気液平衡係数、密度などの物性値を計算する物性計算方法を定義する。ストリームとは、配管に相当し、ユニット間の物質の流れを表現するものである。Visual Modelerでは、プロセスモデル内で実際に存在する成分に絞った部分成分系、及び実際に利用する物質の温度や圧力などに応じ、物性計算方法を複数定義して使い分けることにより、柔軟且つ高速・高精度の物性計算が可能である。

物性と成分リストを定義した後、Visual Modelerが標準で用意するプロセスユニットや計装ユニットが登録されたアイテムメニューから、プロセスに合わせて各ユニットを選び、PFD上に配置する。標準とは別にインストールするオプションユニットやユーザユニット(ユーザ作成のユニット)があれば、ユニットとメニューの登録後プロセスユニット同様に利用できる。

プロセスユニット間に、物質の流れに合わせてストリームを定義する。計器等のユニットが、他のユニットやストリームの情報を参照したり、変更する場合には、

\*1 株式会社オメガシミュレーション

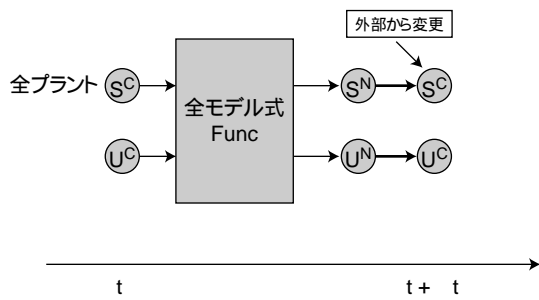


図1 シミュレーション進行

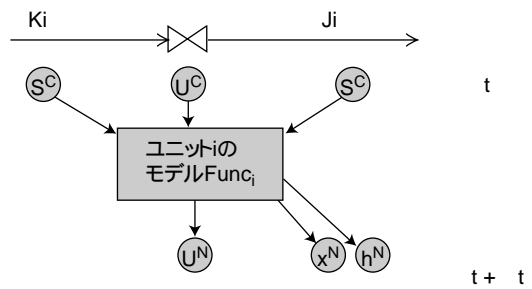


図2 モジュールユニットの機能

信号線で接続する。この信号線接続を利用して、他のプロセスモデルのユニット変数や、OmegaLandの別の機能モジュール内の変数を参照できる。

各ユニットに装置の形状、性能などを定義するパラメータ、物性計算用の系番号などのユニット変数値を入力する。但し、殆どのパラメータにはデフォルト値が用意されているのでこれを利用することができる。

### 3. Visual Modelerによるシミュレーションの仕組み

#### (1) 使用する記号

Visual Modelerでは、ユニット毎に初期化処理やシミュレーション実行時の計算法を、独立に記述するモジュラーアプローチを採用している。

プロセス全体でのシミュレーション概念を、以下の記号を利用して説明する。

$t$  モデル基本周期。プロセス全体のシミュレーションを進行させる基本的な時間刻みである。シミュレーション結果を観察できる最小の時間単位となる。モデル計算の安定性と精度にも影響を与える。ユーザーが指定可能であり、通常のプロセスでは1秒前後が用いられる。

- F ストリームの総モル流量
- P ストリームの圧力
- h ストリームのモルエンタルピ
- x ストリームのモル組成。成分数の要素を持つベクトルである。
- T ストリームの温度
- S ストリーム変数{ F, P, h, x}を併せて表す。
- U ユニット変数。1つのユニット内の全ての変数を総合して表現する。
- Pu ユニット内部圧力( Uの一要素である )
- Fu ユニットにおける流量増減量( Uの一要素である )  
下付き添え字
- i ユニット番号
- j ストリーム番号

- I 全ユニット番号の集合
- J 全ストリーム番号の集合
- Ji ユニットiからのプロダクトストリーム番号の集合
- Ki ユニットiへのフィードストリーム番号の集合  
上付き添え字
- C 現在値
- N 次回値( 現在時刻+  $t$ での値)

#### (2) シミュレーションの進行

シミュレーションは、 $S^C, U^C$ が既知の状態では、 $t$ 毎に $S^N, U^N$ を計算する操作を繰り返すことによって進行する。PFD操作など外部から変更が加わる( 図1 )。

$$S_i^N, U_i^N = \text{Func}( S_i^C, U_i^C )$$

#### (3) モジュールユニットの機能

モジュラーアプローチでは、各ユニットについて、そのユニット内の変数及びプロダクトストリームの組成とエンタルピの次回値は、そのユニット及び隣接するストリームの変数のみから決まる( 図2を参照、図4まで縦軸がモデル時刻 )。

$$x_j^N, h_j^N, U_j^N = \text{Func}( S_i^C, S_{ki}^C, U_i^C ) \quad i \in I$$

式で計算された次回値は  $t$ 後のそれぞれの現在値に等しい( 図3 )。

$$U_i^C(t + t) = U_i^N(t) \quad i \in I$$

$$x_j^C(t + t) = x_j^N(t), h_j^C(t + t) = h_j^N(t) \quad j \in J$$

これらの値は、自身のユニットあるいは、下流のユニットによって次の周期で使用される。

#### (4) 圧流計算

ストリームの流量と圧力は、各ユニット周りの流量バランスと圧力バランスを表す次式から決まる( 図4 )。

$$\text{Func}( F_j^N, P_j^N, U_j^N ) = 0$$

ここで、 $U_j^N$ は 式により各ユニット及び隣接するストリーム変数の現在値の関数である。この式は、プロセス全体に亘る連立方程式として解かれる。

#### (5) ダイナミックシミュレーション実行

以下の順序でVisual Modelerのダイナミックシミュ

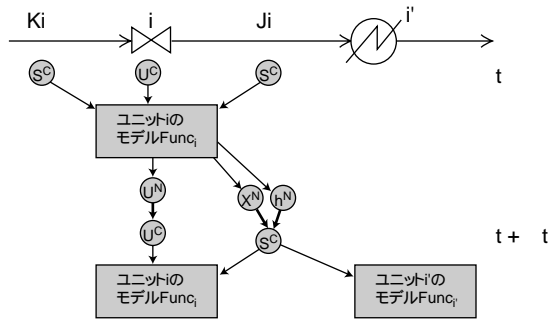


図3 モジュラーユニット間のデータフロー

レーションが行われる。

- (a) 全てのユニットで初期化する。
- (b) 全てのユニットで 式の計算を実行する。
- (c) 全てのユニットで 式をシステムに通知する。
- (d) システムが 式の計算を実行する。
- (e) , 式で次回値を現在値にしてシステム時刻を t 進める。

(a)では、ユニット毎にシミュレーション開始時刻のユニット変数値を、ユニットの設定値から計算する。(b)-(e)は、モデル基本周期ごとに実時間で繰り返される。システムとは、Visual Modelerの実行計算機能である。

#### 4. 圧流計算とユニットモデルとの関係

前項(c)のステップにおいて、各ユニットがシステムに通知する圧流計算の関係式は、ユニットにより異なる。ここでは、気相固定床反応器を例に比較的小さな気相ホールドアップを有するユニットの圧流計算を、システムがどのように処理するか説明する。

気相ホールドアップを有する場合の式は、例えば出入りのストリームが1つつづつの場合、

$$dP/dt = cf \cdot (F_{i1}(P) - F_{i2}(P))$$

の形の微分方程式になる。気相ホールドアップが小さい場合(容器の時定数がモデル基本周期に比較して十分大きくない場合)にオイラー法のような積分計算を行うと、この積分が不安定になることが考えられる。このため、時定数が小さくなることが予想されるユニットでは、右辺を次回値で評価する陰解法を用いる。即ち、

$$P^N = P^C + cf \cdot (F_{i1}^N - F_{i2}^N) \cdot t$$

ここでcfは、ユニット内の変数として現在値から計算される。圧流バランス式は次のようになる。

$$P_{j0}^N = P_{j0}^C + cfdt \cdot (F_{j1}^N - F_{j2}^N - F_{uN}^N)$$

$$P_j^N = P_{j0}^N + dp(F_{j1}^N, U_j^N) \cdot j \cdot (K_i, J_i), j = j_0$$

ここで、式の  $P_{j0}^N$ 、 $cfdt$  及び  $F_{uN}^N$  はユニットモデルで評価される。尚、式中の  $P_{j0}^N$  は  $P_{j0}^C$  と書かれるべきもの

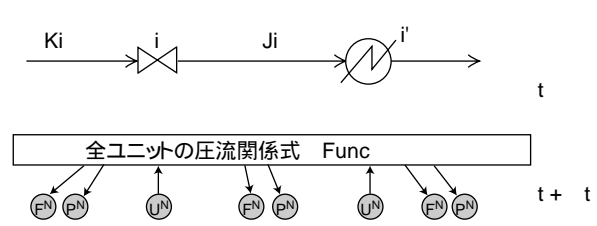


図4 圧流計算

であるが、ユニット側での積分で計算された値を用いてシステムが補正する。これによって、圧流バランス式による積分と、ユニットモデル内部での積分との誤差の蓄積を避けることができる。のdpは内部圧力とストリーム圧力との差を表す。

#### 5. ユニットモデル

Visual Modelerが準備している標準ユニットに加えて、プロセスに特有のプロセスユニットや独自の制御装置などの計装ユニットを、ユーザユニットとして作成することができる。ユニット変数の定義や計算手法を、ユニット毎にテキスト形式のブロックファイルに記述する。ユニットから呼び出される関数は、ライブラリとして別途用意することも可能である。ストリームに接続されるユーザユニットを作成する場合、ユニットは実行内容が記述されたブロックファイル中で、圧流の関係式を設定する関数を利用して、ユニットの状態をシステムに通知する。Visual Modelerでは、こうした仕組みによりユーザユニットを含めた圧流計算が可能となる。

気相反応器のブロックファイルの骨格を示す。( ) で記述した部分は、具体的な処理内容を省略している。

```

STATIC REAL Vol=* "容積[m³]";
STATIC COMP u "成分ホールドアップ[kmol]";
xini=* "初期状態[mol/mol]";
STATIC REAL Hu "熱ホールドアップ[MJ]";
P=101.3 "圧力[kPa]"; T=25 "温度[C]";
cc=0.01 "圧損係数"; Qr "反応熱[MJ/h]";
dr "反応による増加モル量[kmol/h]";
FEED f1; PROD p1; SYSTEM s1=* "系番号";
MODEL{
(初期化处理)
PROC exec C_LANG{ /*実行周期計算(C言語)*/
(変数宣言, 変数現在値読み出し)
/*----- 流入・流出量の計算 -----*/
rev = StrGetTota(" f1 ", &Ff, &Tf, &Pp ) /*流入処理*/
StrGetVapor( sys, " f1 ", rev, &rvf, &hvf, xvf )
frf = Ff*rvf;
rp = StrGetTota(" f1 ", &Fp, &Tp, &Pp ) /*流出処理*/
StrGetVapor( sys, " f1 ", rp, &rvp, &hvp, xvp )
frp = -Fp*rvp;

```

```

for( ii=0; ii<nc; ii++ )f [ii] = frf*xvf [ii]+frp*xvp [ii];
hf = frf*hvf+frp*hvp;
/*----- 物質・熱収支計算 -----*/
for( U=dR=0.0, ii=0; ii<nc; ii++ ){
    u [ii] += ( f [ii]+dr [ii] ) *step;
    if( u [ii]<0.0 ) u [ii] = 0.0;
    U += u [ii]; dR += dr [ii];
}
for( ii=0; ii<nc; ii++ )x [ii] = u [ii]/U;
Hu += ( hf+Qr ) *step; hu = Hu/U;
PhyTcalcV( sys, &T, &Pold, &hu, x ) /*温度計算*/
( 内部圧力Pnew の計算 )
/*----- 圧流バランス指定 -----*/
etap = Pnew/Vol/dv*step;
ComGetRea( " cc ", &cc ) /*ユニット変数読込*/
MdlDefineGroup( WITH_VOL, CONST, -dR, Pnew )
MdlSetupNozzle( " f1 ", VOLUME, etap, 0.0 )
MdlSetupNozzle( " p1 ", QUADRA, cc, 0.0 )
MdlCompleteGroup( 0 )
( 次回値出力, ストリーム変数書き出し )
};}

```

このブロックでは、「MODEL」の前の部分でユニットの変数宣言、初期値設定、ストリーム接続宣言が行われており、プロセスモデルの編集時及び実行時にPFD上で参照や設定が可能である。「PROC」以下は実行周期毎の処理であり、現在値を読み込んで物質・熱収支計算など動特性の計算を行い、「圧流バランス指定」の部分でMdlDefineGroup、MdlSetupNozzle、MdlCompleteGroup関数を呼び出して、ユニットの圧流パラメータをシステムに通知している。ここではQr、drの計算は別途行われていると仮定している。

このようにモジュラーアプローチを採用し、システムが全ユニットの圧流計算を処理している。一方ユーザユニットを含む各ユニットはC言語またはEQUATRAN言語を用いて、独自の動特性や外部接続をそれぞれ独立に作成することができる。

## 6. プロセス例とシミュレーション実行機能

5章の気相反応器を組み込んだメタノール合成プロセスを取り上げる。図5は、簡略化したプロセスモデルの実行画面である。プロセスを構成するユニットは、VV01やTCV01などバルブ、HX01など熱交換器、FA01フラッシュドラム、FT04など各種計測器、FC04などPID制御装置、GB01コンプレッサなどの標準ユニットとユーザユニットであるRA01気相反応器で構成されている。

シミュレーションの実行は、基本モデル周期や、実行時間スケール(実時間)を設定してから「運転」(右向き三角ボタン)で行う。PFDのユニット図形上或は作成アイテムリストから「更新表示変数の選択」を利用すると、図中のT=60.0011などのユニットやひし形で表示されているストリームの変数値をPFD上にリアルタイム表示可能にな

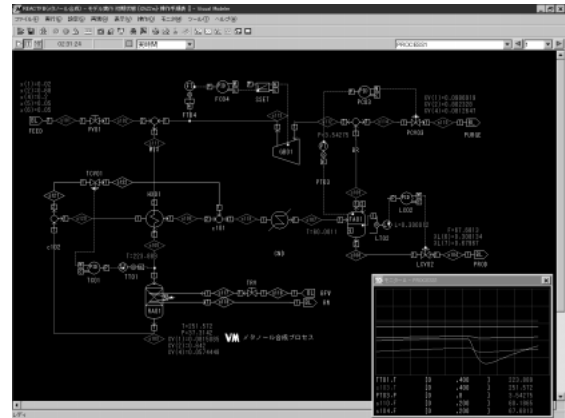


図5 メタノール合成プロセスの実行画面

る。また、「トレンド」機能で同図右下のようなプロセス変数の時間トレンドを作成・表示することができる。ストリームの組成、圧力、流量、温度などを一覧表示するストリーム表、同一ページで表示できない離れたユニット変数値の表示や、変更が可能な監視パネルなどの運転や監視に有効な機能も用意されている。文献3に記述されたOmegaLandグラフィックモジュールを利用することで、より本格的な表示・操作機能を実現できる。

運転状態の一時保存や再現、ファイル保存とファイルからの読み込み機能、及び外部プロセス変数やファイルとの入出力機能が利用可能である。

## 7. おわりに

Visual Modelerは、UNIXシステムとWindowsシステムで動作するダイナミックシミュレータとして開発された。今後は、更にモデル開発及びエンジニアリング効率向上のためのツールの整備、物性値ライブラリの充実と物性計算法の追加によって、多様な物性計算を可能とするなどユーザの要望を実現するように機能向上を進めていきたい。

## 参考文献

- (1) 小口梧郎他, “プラントモデル構築ツールVisual Modeler”, ケミカルエンジニアリング, vol. 39, no. 11, 1994, p. 73-81
- (2) 湯本隆雅, “進化するダイナミックシミュレータVisual Modeler”, 分離技術, vol. 29, no. 6, 1999, p. 23
- (3) 三浦真太郎, 横山克己, “OmegaLand開発コンセプトと機能概要”, 横河技報, vol. 45, no. 1, 2001, p. 63-66

\*2 OmegaLand(株)オメガシミュレーションの登録商標です。

\*3 Visual Modelerは、三井化学株式会社の登録商標です。