

Modelica について

株式会社 電通国際情報サービス
エンジニアリングソリューション事業部
CAEソリューション部

Modelica は 1990 年代後半に仕様が決定し、現在では欧州において広く利用されるようになった「物理モデリング用計算機言語」である。一般に詳細な部分を見ることが多いエンジニアリングツールと異なり、Modelica を用いることによりシステムの視点のシミュレーションが可能になる。本稿では、その歴史、欧州での取り組みを中心にその利用状況、具体的な事例及び技術的な背景について述べる。

1. Modelica の歴史

Elmqvist は Dymola¹ (Dynamic Modeling Language) を 1978 にすでに紹介している。これには方程式ベースの非因果 (4.4 参照) モデリング、再利用のためのモデル型、階層的モデル化のためのサブモデル呼び出しという考え方が含まれている。Andersson, Mattson らは、Omola にオブジェクト指向で方程式ベースのモデリング、クラス概念を導入し 1989 年に発表した。また同年発表された ObjectMath (Fritzson ら) は Mathematica を拡張するハイレベルプログラミング環境及びモデリング言語であり、方程式・代入式の双方が用いられ (4.4 参照) 継承 (4.5 参照) 記号操作などの特徴を有する。

Elmqvist はこれら一連のプログラムの概念を取り入れ、より良い統一の言語体系とするため 1996 年に Modelica Design Group を発足させ、翌 1997 年には言語仕様として Modelica 1.0 が発表された。2000 年には非営利団体として Modelica Association が発足、第 1 回の国際 Modelica 会議が行われた。同会議は 1 年半のサイクルで開催されており、2011 年に第 8 回会議がドイツ・ドレスデンで開催されている。その間、言語仕様も改良され 2010 年に 3.2 がリリースされている。

Modelica 関連の著作としては 2001 年 Tiller が、2004 年に Fritzson が刊行しており、日本では Tiller の訳本が 2003 年古田により出版されている。

¹ 現在市販されているソフトウェア Dymola(表 2 記載) は Dynamic Modeling Laboratory の略である。

表 1 Modelica 関連年表

年	事項
1978	Elmqvistら Dymola を発表
	Omola, ObjectMath などの言語・環境が発表
1996	Modelica Design Group が発足、言語としての一本化開始。
1997	言語仕様 1.0 リリース
2000	Modelica Association 発足 第 1 回国際 Modelica 会議開催
2001	Tiller 著書 Introduction to Physical Modeling with Modelica 刊行 (日本語版刊行は 2003 年)
2002	言語仕様 2.0 リリース
2007	言語仕様 3.0 リリース ITEA2 プロジェクト内のサブプロジェクトとして EuroSysLib プロジェクト がスタート
2010	言語仕様 3.2 リリース
2011	第 8 回国際 Modelica 会議開催

2. Modelica 利用状況

Modelica は発祥地であるヨーロッパを中心に利用が広がっている。

ITEA (Information Technology European Advancement) 2 と呼ばれる欧州全体の情報技術系プロジェクトには Modelica を活用するためのプロジェクトが複数あり、2007 年には EuroSysLib プロジェクトが開始した。同プロジェクトは 2 年 9 か月に及ぶプロジェクトで、19 団体が参加している。代表的な企業としては、Siemens、ABB などがユーザ企業として参画、研究機関としては DLR (ドイツ航空宇宙局)、AIT (オーストリア技術院) などが名を連ねている。この成果物として物理モデルを Modelica で構築する際にその基本となる部品モデルを集めた「ライブラリ」が作られ、利用者の生産性を高める役割を果たしている。産業界では特に自動車分野での利用が顕著である。欧州においても自動車のエコ化に対する要請は強い。自動車のエコ化は、動力源であるエンジンの効率化とともに、複数動力源化 (ハイブリッド化) と電力の効果的な回生、機械エネルギーと電気エネルギー消費の制御に依存する部分が大である。ECU で制御されるエンジンなどの動力源、エアコンなどのエネルギー消費装置では AUTOSAR 規格で制御ソフトが通信・動作できる環境を構築可能になっているが、熱などを含む物理的な動作をモデル化するために Daimler、Volvo、Volkswagen などでは Modelica で制御対象をモデル化する取り組みが行われている。これによりエネルギー消費を抑えるべく制御ソフトが設計されるようになっている。

このほか、エネルギー産業 (Siemens のタービン部門、ABB など)、ビルや地域での冷暖房 (米国 Mitsubishi Electric) から工業用ミシンの開発まで広い分野で活用されている。日本では、トヨタ自動車、マツダ、デンソーなどの事例発表が行われている。トヨタ自動車からは 2002 年の第二回国際 Modelica 会議にてターボつきエンジン事例発表が行われている。1 で述べた Tiller の訳本も杉木及びトヨタテクノサービスが訳者となっている。当初 Modelica は Dymola 上で動作するのみであった。しかし需要増加に伴い、有償のソフ

トウェアやフリーウェアが Modelica を動作させるツールとして提供されるようになってきている（表 2）。

表 2 Modelica ツール

ソフトウェア	開発元	提供形態
Dymola	Dassault Systemes AB	有償
MathModelica	MathCore AB	有償
SimulationX	ITI GmbH	有償
AMESim (一部)	LMS	有償
MapleSim	Maplesoft	有償
Vertex	Deltatheta	有償
Jmodelica	Modelon	無償
OpenModelica	OSMC (Open Source Modelica Consortium)	無償

3. Modelica によるモデリング事例

ここでは Modelica を用いたモデリングの事例を示す。3.以降ではグラフィカル表現でモデルを説明するが、グラフィカルな要素にはそれぞれに対応した文字列として書かれる Modelica ソースコードが必要である。モデル全体を表すソースコードでは各要素がテキスト形式で表現され、要素間の接続もテキストとして表現されている。

3.1 油圧回路

シリンダにより負荷を駆動する油圧回路を含むシステムを考えてみる（図 3.1）。

油圧システムと機械的な駆動システム、位置制御システムから構成される。それぞれの構成は

- ・油圧システム：タンク ポンプ 切替バルブ シリンダ 切替バルブ タンク
- ・駆動システム：支持（固定） シリンダ 負荷
- ・位置制御システム：シリンダ センサー 二値化 方向弁

のようになる。油圧システムには油圧系の、駆動システムには機械運動の、それぞれ別の物理現象が発生する。

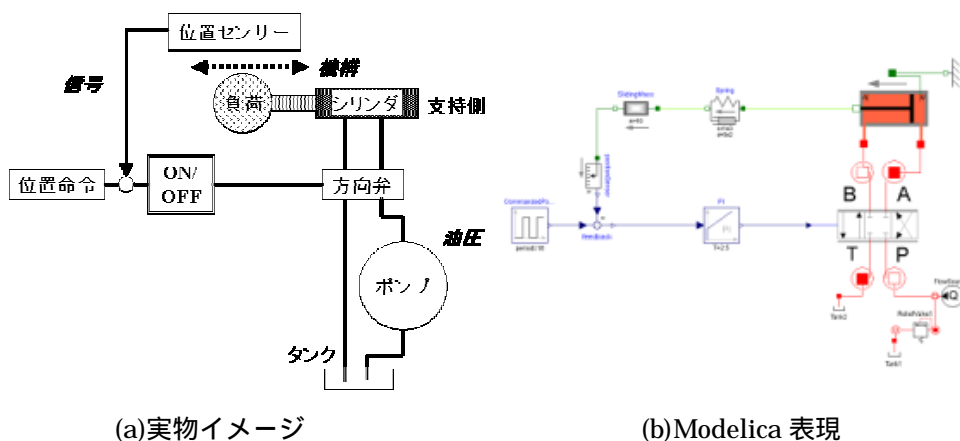
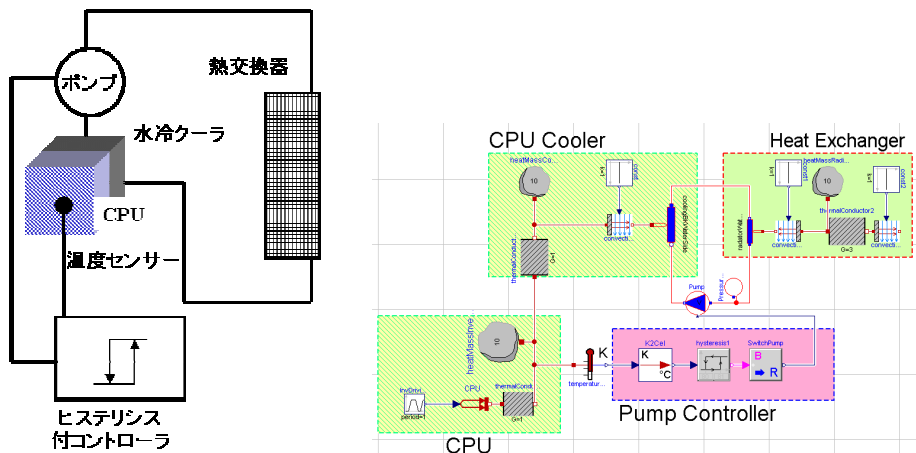


図 3.1 油圧と制御回路

3.2 電気回路と冷却装置

次にCPUの水冷問題を考えてみる（図3.2）。CPUで発熱した熱は液冷のヒートシンクを介して吸収され、熱交換器を通して放熱する。冷却液はポンプで循環するが、消費電力を抑えるために常時通電ではなく、設定温度に達するとオンになり、十分温度が下がるとオフになるようなヒステリシス特性のあるコントローラで温度制御される。

この問題を分解すると水冷管路と熱交換器による放熱、水冷クーラでの吸熱、CPU発熱による温度上昇の計測とポンプの運転制御という問題に分解される。



(a) 実際のモデルイメージ

(b) Modelica 表現されたモデル

図 3.2 CPU クーラ

3.3 エンジン排ガスとターボ機構（副島,2002）

ディーゼルエンジンでは、出力性能向上と排気ガス規制に対応するため、VGT(Variable Geometry Turbo) や EGR (Exhaust Gas Recirculation) システムが導入されることがあった。VGTは排気圧力の上昇を制限するため燃焼ガスの流れを入口部に設けたバーンを用いて調整する。EGRはNO_xを減少させるために、排気ガスをインテークマニフォールドへ還流させる。VGTとEGRが同時に動作することによりインマニ側の圧力が双方の影響を受けてしまう。

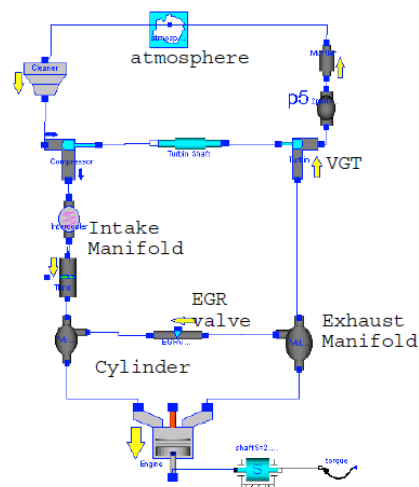


図 3.3 VGT と EGR モデル

これらをシミュレートするために作成されたモデルが図 3.3 のモデルである。このモデルでは、エンジンの燃焼サイクルによるトルク変動や筒内圧力変動は計算していないが、エネルギー保存則、質量保存則とニュートンの運動方程式を用いて現象をモデル化している。VGT はタービンシャフトがコンプレッサーとタービンを機械的に結んでいる。エアクリーナから吸入された空気はコンプレッサーにより圧縮されたあと、インタークーラで冷却され、スロットルを経てインテーク間にフォールドへ送られる。さらにエンジンのシリンダを経てエキゾーストマニフォールド、VGT のタービン、触媒、マフラーを経て排気される。EGR はエキマニとインマニを結んでいる。

ここでは VGT の機械-流体相互作用的な動作、つまりガスの運動量を回転運動に変換し、その回転運動により空気が圧縮される計算と、空気流れ・ガス流れの混合を計算している。このような複雑な動作が簡便に表されているのは、各装置をモデル化した専用のライブラリを用いているためである。

4. 技術的背景

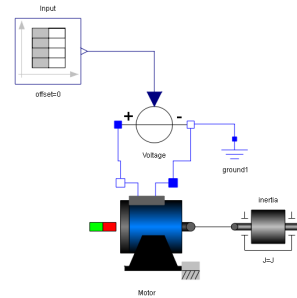
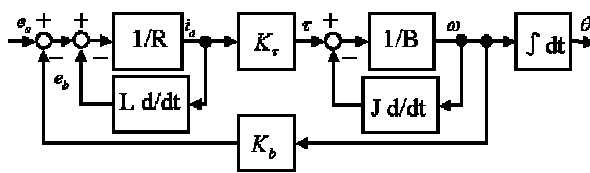
物理現象を評価するために用いられるコンピュータシミュレーションでは、通常、有限要素法（主に構造解析や磁場解析）、境界要素法（主に音場解析）、有限体積法（主に流体解析）などが用いられている。これらは各部品やアセンブリの詳細分布を見る上では有用であるが、システム全体をシミュレートする上では計算時間がかかり過ぎる。問題により局所的な判断ではなく、全体的な動きのシミュレーションが求められる場合には、Modelica によりモデル化することが非常に役に立つ。以下では、Modelica の持つ特徴的な点について述べる。

4.1. 信号モデリングと物理モデリング、グラフィカルモデリング

一般的に用いられているシステム表現手法は、信号の流れをモデリングしたものである²。代表的なものとしてブロック線図が挙げられる。ブロック線図表現（図 4.1(a)）では、ブロック間の接続は矢印線で表される。この線は信号の流れる方向を示すのみで、物理的な意味を持たない。このため物理次元とは無関係に接続ができ、例えば速度と加速度をつないでしまうなどの誤ったモデル化を引き起こしやすい。

Modelica でもブロック線図に似たグラフィカルなモデル化が可能であるのはこれまで見てきたとおりである。ただし接続線は物理的事象のつながりを示す場合と、信号の流れを示す場合を混在することができる（図 4.1(b)）。物理的事象のつながりを表す場合は互いに接続が可能な要素が限定される。例えば左側破線枠内の「バネと集中質量の接続」や「電気抵抗とコンデンサの接続」は可能であるが、右側破線枠内の「バネと抵抗」「集中質量とコンデンサ」は接続できない（図 4.2）。

² データフローモデリングという表現もあるが、数値だけの流れなので信号という表現を用いている。



(a) ブロック線図モデル

(b) Modelica モデル

図 4.1 ブロック線図モデルと Modelica モデル

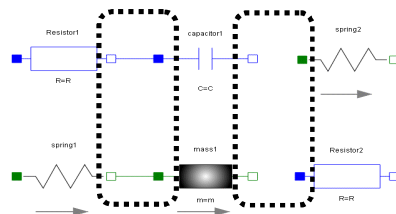


図 4.2 接続可能と不可能

4.2 マルチドメイン性

図 4.1(b)で示したように Modelica では複数の物理現象を同時に扱うことが可能である。無償で標準提供されている「ライブラリ」(MSL : Modelica Standard Library) で表現可能な物理現象を表 4.1 に示す。例えば機械系と電機回路、磁気回路の部品を組み合わせ、一つのモデルとして計算することが可能である。

表 4.1 MSL とドメイン

名称	内容
Mechanical	並進運動、回転運動、3次元運動
Electrical	アナログ回路、デジタル回路
Thermal	熱伝導、熱輻射、熱伝達
Magnetics	磁気回路
Fluid	管路系流体

MSL 以外にも有償提供されているライブラリがある。代表的な有償ライブラリを表 4.2 に示す。

表 4.2 有償ライブラリ

名称	内容
Vehicle Dynamics	自動車の走行シミュレーション用ライブラリ
Power Train	ギアを中心とした自動車駆動系のライブラリ
Smart Electric Drive Alternative Vehicle	電気自動車用ライブラリ
Air Conditioning	(主に)自動車空調用ライブラリ
Hydraulics	油圧系ライブラリ
Pneumatics	空気圧系ライブラリ

4.3 コネクタ

4.1 で述べたように Modelica では要素間の接続の可・不可がある。これは要素種別毎に定義されているコネクタと呼ばれる接続口に物理量が定義されているためである。

MSL において定義されている物理量の例を表 4.3 にまとめる。

表 4.3 コネクタと物理量

	コネクタ正式呼称	物理量（横断変数）	物理量（通過変数）
機械系並進運動	フランジ	位置・座標	力
電気回路	ピン	電圧	電流
熱	ポート	温度	熱流

物理量（通過変数）は接続されているコネクタの合計値がゼロとなる。

物理量（横断変数）は接続されているコネクタの値が等しくなる。

4.4 因果的と非因果的

一般的なプログラミング言語は演算において代入文を用いる。

$$X=A+B \quad (1)$$

は X に A と B の合計値を代入することを示している。Modelica にもこのような代入文の形式は存在するが、=（等号）は代入ではなく文字通り「等しいこと」を示す。従って、(1) 式は

$$X-A=B \quad (1)' \quad \text{または} \quad A=X-B \quad (1)''$$

と記述しても計算結果は同じになる。

未知数を解くのに必要十分な数の等式をソースコード内に記述することで、Modelica 実行環境（ツール）は、これらの等式を自動的に連立方程式として展開しシミュレーションし数値解が導出する。利用者側は方程式を代入式に解く必要がない。方程式の中に時間微分が含まれるような場合、例えば速度（v）を時間微分して加速度を用いる必要がある場合、既存の変数 v に加えて新たに加速度を示す変数 a を導入し

$$dv/dt=a \quad (2)$$

のように表現し等式を増やすことができる。これにより等式の数を変数と等しくすることができるので解を求めることができる。さらに時間のファクターを導入することになるので動的な問題も解けるようになる。

ここで(1)式を再度見てみる。代入文である場合には、A、B がともに入力、X を出力と考えることができる。このように出力と入力に明確に区別されているものを因果的（causal）と呼ぶことにしている。一方 Modelica で扱っているように等式とみなすと、入力と出力が明確に区別されるわけではなく、相互に影響を与え合う関係であることになる。これを非因果的（acausal）と呼ぶ。

非因果的な取扱いが可能になることで、次のような現象が表現できる。

「押すと押し返される」

「モータを外力により回転させると発電できる」

4.1 で述べた信号モデリングの手法では図 4.1(a)のようにループを作成することにより現象をモデル化できるが、実物として存在する物理的な接続とは異なる接続になるため、これから現物であるモータを認識するのは非常に難しい。また別の機構などを付加するとループ構成を含めてモデルをはじめから見直す必要があり再利用するのが困難である。

4.5 クラスと継承

Modelica はオブジェクト指向のモデル化言語であり「クラス」の概念が導入されている。またオブジェクト指向言語の特徴として「継承」が可能になっている。継承では、もともになる要素(クラスの一つ)を発展させて項目の追加が可能となる。継承を利用することにより、ベースになる要素の改善改良を行った場合、その改修内容は継承したすべての要素に波及するため、一つ一つに対して修正を加えていく必要がなく保守性も向上する³。

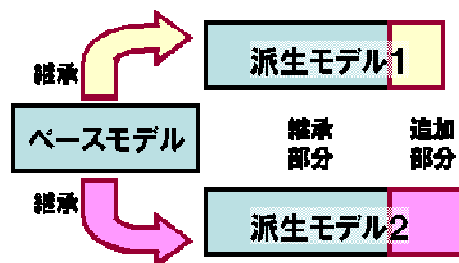


図 4.3 継承

4.6 ソースコードとオープン性

Modelica の言語仕様は公開されており誰でも自由に入手できる。また MSL のソースコードは全て開示されている。したがって、継承機能などを用いてユーザは自由にかつ効率的に物理モデルを定義することができる。Modelica に似たプログラムは他にも存在しているが、その多くは内部ロジックを公開していない。このためユーザ自身が拡張する場合に制限が多く、その結果として費用負担が大きくなる傾向がある。

4.7 制御との関わり

MSL にも伝達関数や PID 制御などのブロックが用意されている。これらを用いて Modelica を使ったモデルの中で制御対象と制御ロジックの双方を表現しシミュレーションを行うことも可能である。しかし多くの場合、大規模な制御ロジックは MATLAB/Simulink で作成されることが多い。これは作成した制御ロジックを実際のプログラムとして組込む必要があるからである。Modelica のツールには、Modelica で作成した物理モデルを MATLAB/Simulink に渡す機能を有しているものが多い。物理モデルは Simulink の S-function として引き渡されていたが、2 で紹介した AUTOSAR との接続規格に準拠した FMU(Functional Mockup Unit)として渡す形式に代わりつつある。

また Modelica のツールには、モデルを C ソースコードで書き出す機能を有するものがある。書き出された C ソースコードは、HILS ハードウェア上で動作する実行モジュールとして

³ 継承の関係が正しく把握されていることが前提になる。逆に管理の負荷がかかるという意見もある。

利用される。

5. 終わりに

Modelica は比較的新しい「物理モデリング用計算機言語」で、本稿ではその歴史から技術的な背景までを説明してきた。欧州では一層の活用が見込まれており、計算機システムと実際の物理システムを結びつけたシステム（Cyber Physical System）を表現する上で重要な役割を果たす言語になるものと思われる。対抗軸として米国でも私企業ベースのツールで同様な取り組みが行われているが、今後はそれぞれの動向を注視していく必要があると考える。

（2011年9月5日）

文中の会社名・ソフトウェア名は、多くの場合 各社の商標または登録商標です。
この文書の中で使用されている Modelica の図などは、the Modelica License2 に基づいて記載されている場合があります。
the Modelica License2 については
<https://www.modelica.org/licenses/ModelicaLicense2>
をご参照下さい。