組込みシステムのソフトウェアのモデル設計に関して

2018SE055 長岡楓己　　2018SE036 黒川誠史
指導教員：横山哲郎

1　はじめに

今日多くの製品，機器にソフトウェアが搭載されている．その例として，スマートフォンや道路の交差点の信号機，あるいは自動車などである．

自動車には ECU(Electronic Control Unit) という自動車のエンジンを電気的な補助装置で制御することに使用するマイクロコンピュータを電子回路を用いて制御する装置にもソフトウェアが搭載されていることが挙げられる．

例で挙げた ECU のマイクロコンピュータは一般的に図 1 のように，エンジンと同じように車内に搭載されている．これらのような製品に搭載されたソフトウェアは特に，組み込みソフトウェアと呼ばれる．さらに，それらのような製品，機器のシステムの動作に影響を与える組込みソフトウェアは組み込みシステムと呼ばれる．すなわち例で挙げたものそのもののことである．

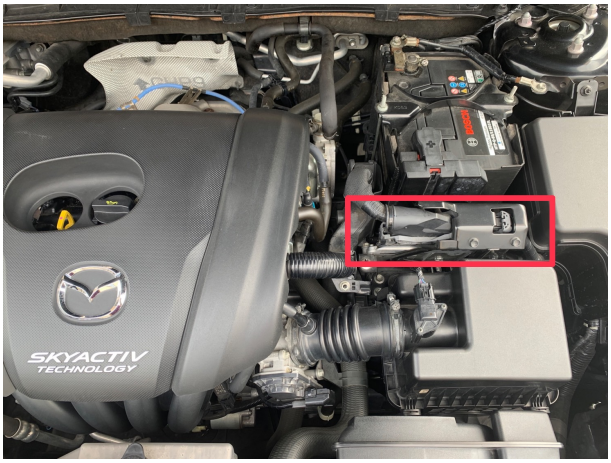

図 1　ECU のマイクロコンピュータ (赤枠内)

1.1　背景

組込みシステムを設計するために，理解しておくことが [1] で簡易的に説明されている．

以下小小節ではその理解しておくことやその用語を述べる．

1.1.1　CPS

cyber-physical systems の略で，センサから計測された情報をコンピュータ間で処理し，その処理した情報をもとにアクチュエータの動作に影響を及ぼすシステムを指す．

組込みシステムは CPS である．

[1] によると，CPS は IT の進化に役に立つと考えられている．実際に，DX(Digital Transformation) という概念が一般に推進されていることから，その動作元であるソフトウェア設計の理解が重要であると考える．

1.1.2　[1] による組込みシステムの設計

[1] では，CPS である組込みシステムを作成するには，大まかに 3 つの行程からなる．モデリング，設計，分析である．

モデリングの行程では，システムの仕様を定め，システムがもつ特性を設計することである．

デザインの行程ではモデリングの行程で作成したモデルの構造化を行う．どのようにシステムを実行するか手順を定める．

分析の行程では，デザインの行程で定めた実行の手順を実行し，システムが目的の機能を成すか分析する．

通常，モデリングから考案し，解決したい問題を理解し組み込みシステムを作成する．

1.1.3　モデリング

モデリングとは実世界の「もの，こと」についての特徴や，これらを対象とした適切な性質を抽出することによって簡略化された抽象的なモデルを作成することである．
モデリングの具体例を 2 つ挙げる．

1. システムの動作に関する微分方程式をたて，ソフトウェアモデリングおよびシミュレーションツールでアクタの関係や役割を決める．
2. アクタの状態遷移のモデル図を書く．

1.1.4　連続ダイナミクス

モータの回転など，システムの連続的な動作は連続ダイナミクスと呼ばれる．

特性として，実世界の空間での物の動きは 6 つの度合い，値で表すことができる．

3 つの軸 xyz からなる空間を実世界の空間としたとき，
位置 xyz の座標．
ロール：x 軸を中心とした回転．
ヨー：y 軸を中心とした回転．
ピッチ：z 軸を中心とした回転．

システムの動作が，線形性を有するものや時間によるモデルそのものの変化が無いものでのモデルは一般に，微分方程式が使用される．1 種類の微分方程式たちでモデルを表現できない場合は，モーダルモデルを使用する．モーダルモデルは瞬間的 (離散的) な動作と連続的な要素を併せ持つ動作をモデル化することが得意である．

1.1.5　離散ダイナミクス

離散システムとは，一連の離散ステップで動作し，離散ダイナミクスを所持している．
図 2 では，Arrival Detector は車が到着した際にイベントを生成し，Departure Detector は車が出発した際にイベントを生成する．Counter は，初期値 i から開始し，実行カウントを維持する．カウントが変更されるたびに出力イベントが生成される．
入力が離散的であると仮定すると，Counter は一連の入力イベントのそれぞれに順番に反応する．Counter への入力は特定の場合にイベントが発生する場合と発生しない場合がある離散信号のペアである．出力も離散信号であり，入力が存在しない場合は Counter は何もしない．入力が存在する場合のみ動作する必要がある．したがって，離散的ダイナミクスが存在する．

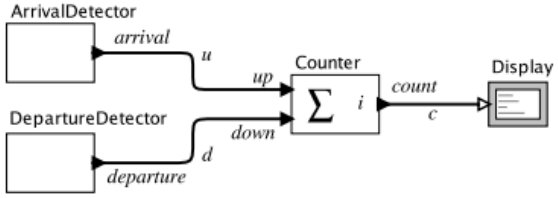

図 2　駐車場に出入りする車の数をカウントするシステムのモデル例

1.1.6　状態の概念

状態とは，システムが入力に対してどのように反応するかに影響する．
現在または未来の入力に対するシステムの反応に影響を与える，過去に関するすべてのエンコーディングとして定義する．
図 3 のアクターには状態があり，出力がいつでも同じになる．

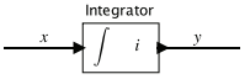

図 3　インテグレーターアイコン

1.1.7　有限状態マシン

離散ダイナミクスのシステムのモデルであるステートマシンの中で，特にモデルの状態 (ステート) が有限であるものを，有限状態マシン (FSM) と呼ぶ．
有限状態マシンでは，モデル図を書いてモデルの表現する．図 4 では温度調節器のモデル図を示した．冷却を初期状態とし，入力の温度が摂氏 18 度以下であれば，加熱の状態へ遷移する．入力の温度が摂氏 22 度以上であれば，冷却の状態へ遷移する．以後，これらの条件で状態遷移を続ける．

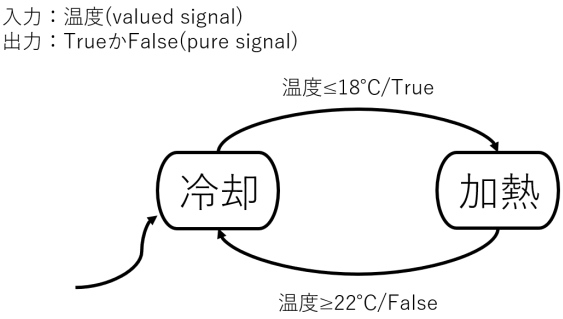

図 4　温度調節器のモデル図

ステートマシンでは状態を遷移をするか否か，出力をどのアクタへ生成するかを定める必要がある．
ステートマシンには 2 つの特性がある．
決定性:等しい入力に対して, 常に出力が等しい．
受容性:常に入力に反応する準備ができており, どの状態でも実行可能である．つまり入力が 1 つ定まればそれに対応した出力が 1 つ決まる．

1.1.8　ハイブリッドシステム

微分方程式とアクタモデルでモデル化される連続ダイナミクスとステートマシンでモデル化される離散ダイナミクス，この 2 つを併せ持つシステムをハイブリッドシステムと呼ぶ．
ハイブリッドシステムの例として，温度調節器の例が挙げられる．図 4 のモデル図は，入力は温度を数値化したものでありデータは連続的，出力は加熱か冷却であり純粋な信号で離散的である．

1.1.9　センサとアクチュエータのモデル

まずはセンサとアクチュエータ，それぞれの一般的な説明から述べる．センサとは，物理的な変量を測定しそれをデータに変換する機器である．物理的な変量を観測するために，物理的な変量によって影響させられた電圧の変化を計測する．センサが計測した物理的な変量を，連続データへ変換するセンサをアナログセンサ，連続データへ変換するセンサをデジタルセンサと呼ぶ．データの精度は，デジタルセンサがアナログセンサより劣る．アナログデータをデジタルデータへ量子化する機器をアナログ-デジタル変換器（ADC）と呼ぶ．
アクチュエータとは，入力されたデータ (数値) に従い，電圧を変化させ物理的な変量を変化させる機器である．量子化されたデジタルデータをアナログデータへ変換する機器をデジタル-アナログコンバータ（DAC）と呼ぶ．
物理的な変量には，2 つの種類がある．環境そのものによって影響される重力等の加速度と，環境そのものによって影響される加速度と物体そのものによる加速度を合わせたものがある．前者は特に固有加速度と呼ばれる．

センサが計測したい物理的な変量について，加速度はバネを利用する．計測した加速度を利用して位置求める技術があるが，ドリフトと呼ばれる誤差が大きい．この誤差を補う他の計測には，GPS,WiFi,Bluetooth 等で電波強度を用いる方法がある．

回転角を求めるには，角加速度をジャイロスコープと呼ばれる光学デバイスを用いた技術がある．

これらを計測するセンサは慣性測定ユニット (IMU) と呼ばれる．

アクチュエータの例として LED がある．LED は点灯か消灯か，あるいは電流を流すか否か制御される．この制御はデジタル I/O ピンと呼ばれる純粋な信号を用いることから離散ダイナミクスの特性がある．

他の例として，モータがある．モータはどれくらい回転するか，あるいは必要なトルクに対応した電圧をモータに加えることを制御される．　この制御は，値を表す信号を用いることから，連続ダイナミクスの特性がある．

通常，組込みシステムで使用するセンサやアクチュエータは API が用意される．

1.1.10　組込みプロセッサ

組込みプロセッサは，通常専用の機能を持たせるように設計される．例えば，自動車のエンジンを制御するためのみに設計される組込みプロセッサがある．

プロセッサの種類には，マイクロプロセッサや，他にも非常に電力を消費するためエネルギー制約のある組込みシステムには不適であるグラフィックプロセッサがある．

プロセッサが行う処理には, 並列処理 (Parallelism) と並行処理 (Concurrency) がある．

並列処理は, 物理的に同時に実行される. 複数のプロセッサで異なる処理をする．プロセッサごとの処理が単純化される．並行処理は，概念的に同時に実行される．実行処理が高速化される．

並行でない処理をする実行する命令型言語 C では，一連の命令を定める必要があるが，Java ではスレッドと呼ばれる並行処理ができるライブラリがサポートされている．コンパイラがプログラム内の依存関係を分析 (データフロー分析) し, 並列コードを生成する．

オンザフライとは，ハードディスクにデータをリアルタイムに変換しながら書き込む方式である．

組込みシステムは複数のセンサ，アクチュエータを同時に制御する必要があり，制御を高速化すればいいわけではない．例えばエンジンの点火などは適切なタイミングで制御する必要がある．

実行のポインタの指す場所を変えて別の命令を実行させ，これ自体は何もしない，あるいはこれ以外は何も役割を果たさない動作をする命令をノーオペレーション命令と呼ぶ．

命令を実行する工程を分割し，各段階を重ねて処理する仕組みをパイプラインと呼ぶ．

1.1.11　PtolemyⅡ

[2] では，CPS を作成するために理解すべきことが [1] による説明に加え，さらに詳細に説明されている．ここでは，シミュレータが紹介されている．そのシミュレータは「PtolemyⅡ」である．さらなる「PtolemyⅡ」についての紹介は，「https://ptolemy.berkeley.edu/index.htm」にて．

1.2　用語の説明

以下に本研究で必要となりうる概念，用語を述べる．

IMU：inertial measurement Unit 慣性計測装置　運動を司る 3 軸の速度 (または角速度) と加速度 (または各加速度) を測る装置.

インターフェース：個々のシステムの間で，仲立ちとなるもの (インターフェースの例：端子の規格).

キャッシュ：読み書きが高速な記憶装置に 1 度利用したデータを保存しておき，次回の利用はそこから読み出して処理を高速化する仕組み.

ステートマシン：状態を定義し処理を切り替えるような構造のソフトウェア.

モーダルモデル：CPS などシステムの特性を表現したモデルであり，1 つのシステムを複数に場合 (モード) 分けしたモデル．あるいはそのモードを指す.

トルク：回転力を表す．一般的に用られる単位は N・m である.

pure signal：純粋な信号 (0 か 1 のみ).

valued signal：pure signal に対して反対の意味合いで用いられる，値を表す信号 (実数の値など).

しきい値:境界の値. 例えばセンサが観測できる最小限の物理量，またはその値.

インターリーブ:データの送受信を行うとき, 1 つのデータを分散すること.

2　目的

本研究では [1] で説明される組込みソフトウェアの設計手法について理解することを研究の目的とする．

実際にシミュレーションツール「PtolemyⅡ」を用い，CPS をモデリングしさらなる設計の理解を深めることを研究の最終的な目的とする．

[1] による具体的な設計手法は以下の通りである．

(1)CPS のモデリング

(2) モデルのデザイン

(3) 組込みシステムの分析

本研究では「(1)CPS のモデリング」に焦点をあてる．

3　手法

実際に [1]，[2]，[3] で紹介された手法やモデルを倣い，PtolemyⅡ を用いて CPS のモデルを作成する．

何か組込みシステムの要求される役割を考える．そのシステムの役割を連続ダイナミクスと，離散ダイナミクスとの 2 つの特性に分ける．これらをそれぞれモデル化する．

このモデルから，連続ダイナミクスと離散ダイナミクスとを橋渡しする処理をモーダルモデルを用いモデル化する．最終的に，作成したモデルを PtolemyⅡ の環境下で実行し，システムの動作をシミュレートする．

4　結果・現状

現時点での成果は以下の通り．

[1] で説明される組込みシステムの設計手法の確認をした．

PtolemyⅡ の実行環境を構築し (図 5)，物理システム「球の落下，跳ねる動作」の例を実行 (図 6) した．

以下に示した図 5，図 6 では球を地面に落下するモデル (あるいはその作成画面)，シミュレーション結果である．物理システムを PtolemyⅡ [2] 上で再現

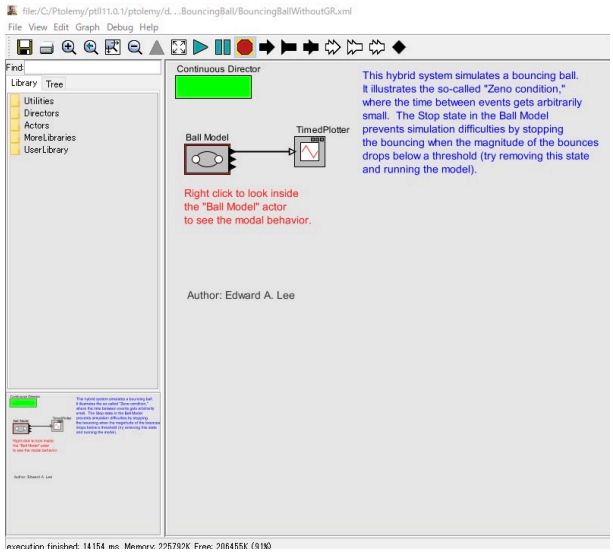

図 5　PtolemyⅡ:モデル作成の例

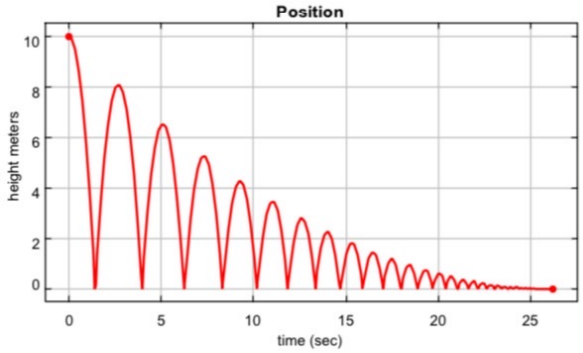

図 6　PtolemyⅡ:シミュレーション例

5　おわりに

5.1　今後の課題

・組込みシステムの設計について [2] 等文献調査を継続し，実際に CPS のモデリングを行っていく．

・PtolemyⅡ の使い方を理解する．

・[3] でさらなる，システム設計の理解を深める．

参考文献

[1] Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, MIT Press, ISBN 978-0-262-53381-2, 2017.

[2] Jeff C. Jensen, Edward A. Lee, and Sanjit A. Seshia, An Introductory Lab in Embedded and Cyber-Physical Systems, http://LeeSeshia.org/lab, First Edition v1.70, 2015.

[3] Claudius Ptolemaeus, Editor, System Design, Modeling, and Simulation Using Ptolemy II, Ptolemy.org, 2014.