Vol.2019-IS-147 No.7
2019/3/7

「話題沸騰ポット」の「ふた」はクラスなのか –学習の階型論によるモデリング試論–

児玉公信[†1]

概要：SESSAME WG2 の「話題沸騰ポット」の分析モデルには疑問がある．そのモデルでは，「ふた」や「給湯口」がクラスとなっている．「ポット」というシステムにおいて，「ふた」や「給湯口」はどれほどのデータや責務を持っているというのだろうか．経験的に，このモデルは少なくとも二つの目的を混合していると感じる．一つはポットの制御システムの設計，もう一つはポットの物理的構造の設計，すなわち部品表である．従来の部品表のモデルは「品目」クラス間の再帰関連で記述され，そのインスタンス群は木構造となる．ここでは，「ふた」や「給湯口」は「品目」クラスのインスタンスとされる．しかし，製品が多仕様化する現代では「ふた」や「給湯口」などの部品のバリエーションが多数あり，それゆえその間のリンクには明確な制約が必要となる．これを記述する部品表は Fowler のいう分析モデルの知識レベルにあるが，これまでデータモデルの視点でしか議論されないできた．本報告では，こうした多仕様の部品表を記述するためのモデリング要素としてステレオタイプ<<item>>を提案し，「話題沸騰ポット」の多仕様版のモデルを記述してみる．この作業を通して，制御システムとのモデルの分離を試みる．

キーワード：組込みソフトウェア，分析モデル，知識レベル，stereotype

Is a "Lid" of the "Hot Topic Water Boiler and Warmer" a Class? - A Tentative Study on Modeling by Logical Category of Learning -

KIMINOBU KODAMA[†1]

Abstract: I have a little discomfort in the analysis model of SESSAME WG 2's the "Hot Topic Water Boiler and Warmer." In the model, "Lid" and "Water Inlet" are classes. In the system called "water boiler and warmer," what data and responsibilities do "Lid" and "Water Inlet" have? Empirically it seems that the model jumbles up at least two objectives. One is designing the control system and the other is designing the physical structure, i.e., the BOM (Bill of Materials). The conventional BOM model is described as a recursive association among "Item" class and itself, and the instances make tree structures. Here, "Lid" and "Water Inlet" are instances of "Item" class. However, in modern BOM for multi-specification and multi-optional products, it requires many kinds of "Lid" and "Water Inlet". And the links between them require clear constraints. Essentially, BOM is at the knowledge level of Fowler's analysis model and only has been discussed at the data-modeling point of view so far. This report proposes a stereotype <<item>> as a modeling component to describe such a multi-specification BOM, and then tries to describe a model of a multi-specification version of a "hot boiling pot" using it. And also tries to separate a model of control system.

Keywords: embedded software, analysis model, knowledge level, stereotype

1. はじめに

オブジェクト指向は「現実世界をそのままソフトウェアに表現する技術である」という誤解は根強く，むしろ有害ですらある[1,2]．Meyer[3]の定義によると，オブジェクト指向設計とは「すべてのシステムまたはサブシステムが扱うオブジェクト(対象，物)をもとにアーキテクチャを構築する手法」あるいは「構造化された抽象データ型の集合として，ソフトウェアシステムを構築すること」であるとされる．クラスは，ここでいう抽象データ型の一つの表現形式であり,「共通の特性によって特徴づけられるデータ構造の集合を表す」．実際は，現実世界をそのままではなく，分析者はそこで扱うべき事柄を解釈して，それらを適切なクラスとして設計し，クラス間の相互作用としてソフトウェアを設計する．オブジェクト指向設計では，このクラスとクラス間の相互作用の決定が本質的な設計作業となる．モデルはそのための設計図面として重要な役割をもっている．

こうした観点から筆者らは，SESSAME WG2[4]による組込みソフトウェア向けのオブジェクト指向モデリングの教科書を標榜する「話題沸騰ポット」の分析モデルにかねてから違和感を持っていた[5]．そのモデル図にあるポットの「ふた」や「給湯口」はクラスなのか，現実世界をそのままソフトウェアとして表現しようとしている痕跡ではないのかとの疑問は解消していない．しかし，多仕様の製品の部品表システム（Bill of Materials; BOM）を検討する中で，この違和感を解決する糸口を見つけたので，クラス設計の問題と併せて報告する．

2. 「話題沸騰ポット」とは

まず，文献[4]の「話題沸騰ポット」の概略を述べる．

†1 (株)情報システム総研
Information Systems Institute, Ltd.

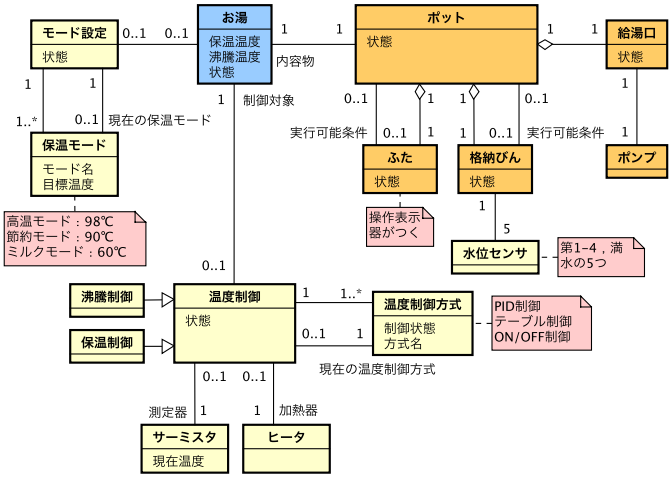

図 1　「話題沸騰ポット」の分析モデル

話題沸騰ポットは，SESSAME WG2 がまとめた「組込みソフトウェア開発のための教科書」の中の例題として使われている想像上の一般家庭用のポット型電気湯沸かし器兼保温器である．これは，ポット内の水を沸かす，保温する，給湯するなどの機能をもつ．例題では，これらの動作を制御するシステムについて，要求モデリング，分析モデリングから設計モデリングまでを実際に行いながら解説を加えるものとなっている．構造化モデリング版[6]とオブジェクト指向モデリング版[4]が提供されている．

2.1 分析モデル

文献[4]にあるオブジェクト指向の分析モデルを図 1 に示す．このモデルは，筆者により操作区画を省略し，必要に応じて注釈を付けてある．

2.1.1 モデルに対する疑問

この例題の主眼は，制御システムの設計にある．このモデルに対する疑問は，「ふた」「格納びん」「給湯口」「お湯」がクラスとされ，しかもそれらは「ポット」と多重度 1 対 1 で関連しているが，このようなクラス構造は組込みソフトウェアにおいては妥当であると言えるのだろうか，そしてそれはなぜかである．

文献[4]の定義によると，「オブジェクト：実世界に存在する認識可能な『もの』」，「クラス：ある共通の特徴を持ったオブジェクト群を抽象化した『型』」であり，「インスタンスは実際に存在し，実際に動作し，属性に具体的な値を持つ『実体』です」とされる．そして，カタログに掲載されているポットの型式と仕様がクラスであり，製造番号を持った個々のポットがインスタンスであると書かれている．このような説明がなされるのは，オブジェクトは実世界に存在する「モノ」であるとする根強い誤解，そしてカタログはクラスで，実体はそのインスタンスであるとする誤解を反映しているからではないだろうか．

2.1.2 真性実世界モデル

磯田[1]は，実世界に存在する「モノ」をクラスとして扱うモデリング手法を「真性実世界モデル」と呼び，その用途は①実世界の問題の理解を容易化する（ソフトウェアとしての実装を考えない），②実世界を模擬実行するソフトウェアを開発するであるとした．これに対する「疑似実世界モデル」は③実世界の業務を自動化するソフトウェアを開発するために用いられ，そのソフトウェアが処理対象とする事物のみをクラス化する．分析者はモデル化手法を混同することなく，用途によってモデル化手法を使い分けるべきであるとした．

2.1.3 組織活動の認識モデル

筆者が考える情報システムの分析モデリングは，必ずしも業務自動化ソフトウェアの開発を前提とするのではなく，組織活動の目的をより円滑に（速く，安全に，誤りなく，容易に）達成できるよう支援するために，組織活動そのものをモデル化する．そこでは，組織マネジャ（あるいはその代理人としての分析設計者）の関心に基づく実世界の解釈[7]が反映される．そうした認識モデルが扱うのは，実世界の「モノ」ではなくて，事物についての理解や情報である．次の図書館業務における「本」の例で考えてみよう．ここでは，本の購入，廃棄，貸出，返却などの移動と，本に関する情報に関心が向く．また，後に述べる部品表のモデルにしても，これらは決して，実世界の模擬実行ではなくて，ソフトウェアをもシステムの要素とする実世界の業務そのものとみなしている．

2.1.4 現物と種類

図書館で扱う本の情報には，いわゆる書誌情報としての「本(Book)」と，1 冊ずつの印刷された「本(Copy)」とがある[8]．前者は実世界にモノとして存在しない「知識」の集合であり，カタログあるいは蔵書目録として外部表現される．後者は実世界にモノとして存在し，手に触れることができ，重さを持つ「本」の集合である．このような状況は図 2 のようにモデル化される．図 2 では Book としての本を「本(種類)」と表記している．このインスタンスは「UML モデリング入門：本(種類)」や「ホモ・デウス 上：本(種類)」である．また，Copy としての本は「本(現物)」と表現している．図書館のシステムなので，このインスタンスは蔵書として扱われ，一つひとつに番号がつけられて識別される．たとえば「12256：本(現物)」，「12257：本(現物)」であり，それぞれがその書誌情報である「ホモ・デウ

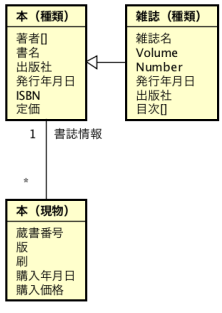

図 2　図書館システムの「本」の部分の分析モデル

ス 上：本(種類)」にリンクする．「本(現物)」が属性に購入価格を持っている理由は，ここでは定価どおりに買わないことが想定されているからである．いずれにしろ，現物の集合はカタログではない．

もし，図 1 の分析モデルの「ポット」クラスがカタログとしての「ポット」だとすれば，その属性には少なくとも「型式」「仕様記述」「定価」などが必要であり，逆に現物の集合としての「ポット」だとすれば，その属性には「製造番号」「購入日付」などが必要なはずである．にもかかわらず，「ポット」クラスの属性には「状態」だけがある．世界のどこかに存在する一つひとつの現物の「ポット」の「状態」をリアルタイムに誰か（おそらくメーカ）が把握すると述べている(今なら把握することは可能だが)．

2.2 知識レベルとしての部品表

実は，種類と現物の区別は，分析モデリングにおいて頻繁に発生する．図 2 において「○○種類」と記したクラスを Odell[9]は○○についてのメタモデルと呼び，Fowler [10]は「知識レベル(Knowledge Level)」と呼んだ．これを明示するために，当該クラスのロールを「Power Type(べき型)」を表記することもある．一方，「○○現物」としたクラスは相対的に「操作レベル(Operational Level)」と呼ばれる．

知識レベルのクラスは操作レベルのクラスのインスタンス生成から消滅までを制約する．たとえば，「本(種類)」のインスタンスが存在しなければ「本(現物)」のインスタンスは生成できないことにするとか，定価より高い購入価格でのインスタンス生成は認めないなどである．このような制約ルールの実装方法にはいくつかある（かつては「○○マスタ」として持ち，その内容を操作レベルのプログラムで解釈していた）が，分析モデルでは知識レベルのインスタンスの属性に記録され，知識レベルで解釈されるものとする．分析モデリングにおいては，知識レベルと操作レベルを分離して表現することにより，情報とその実体を混同した概念化を避けることができる．

2.2.1 一般的な部品表のモデル

知識レベルのモデルの例として，製造業で使われる部品表システムを取り上げる．部品表は製品を生産するための部品構造の制約を表現するモデルである．ここで使用される製品や部品は知識としてのそれであり，現実世界に存在する製品や部品のことではない．一般的な部品表のモデル[11]を図 3 に示す．

部品表において，製品や部品は製造工程における相対的なロールなので一般化して「Item（品目)」とする．製品側の品目（親品目ともいう）と部品側の品目（子品目ともいう）の関係を，図 3 の左のモデルでは再帰関連とし，それぞれに product（あるいは'prod'と表記)，part なるロールを付し，多重度はそれぞれ 0..1，0 以上とする．これに付した{hierarchy}は部品構造がループしないこととする制約ルールである．また，この再帰関連には関連クラス「qtty（員数)」を付けて，製品を 1 単位製造するに当たって必要とする部品点数（員数または所要量）を指定する．

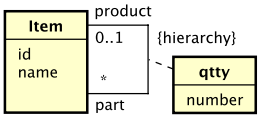

図 3　一般的な部品表のモデル

2.2.2 話題沸騰ポットの部品表

この一般的な部品表モデルに基づいて「話題沸騰ポット」の品目構成を想定して部品表のインスタンス図aを図 4 に示した．員数は水位センサを除いてすべて 1 である．

このモデルは，すべての Item は同じ属性を持つことを前提としている．従来のように，クラスを RDB のテーブルに対応するように実装すると仮定すると，各品目は Item テーブルの行に対応し，それらは id（品目コード）と name からなる．このモデルは id で親子関係を表している．

しかし，製品が多仕様化してくると品目の仕様間での制約（たとえば，「親品目 A の仕様『色』の値が『赤』なら子品目 B の仕様『サイズ』の値は『2』で員数は 3 でなければならない」など）を記述したくなる．これを従来手法で実現しようとして，たとえば「仕様コード」を設けることもあった[12]．しかし，多様化し続ける仕様を固定長で符号化する手法は当然破綻する．これを実際に可能にするにはちょっとした工夫が必要になる．話題沸騰ポットからはいったん逸れるが，知識レベルのモデリングの例として紹介する．

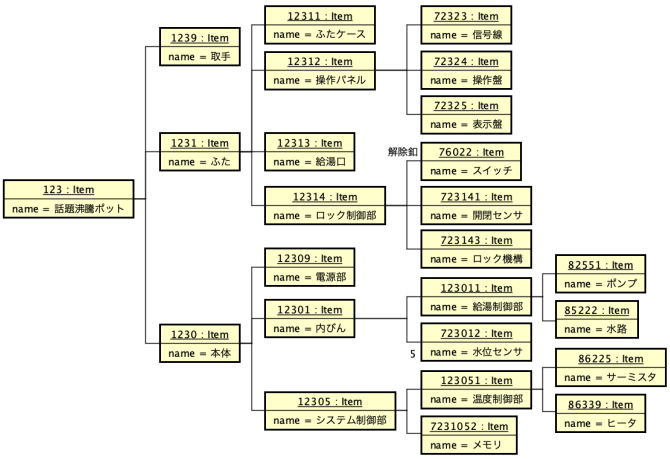

図 4　「話題沸騰ポット」の部品表（インスタンス図）

3. 多仕様製品のための部品表

多仕様化の鍵は「品目ごとに異なる仕様をどのように扱うか」にある．

a UML のオブジェクト図の記法を用いてインスタンスの構造を記述する．

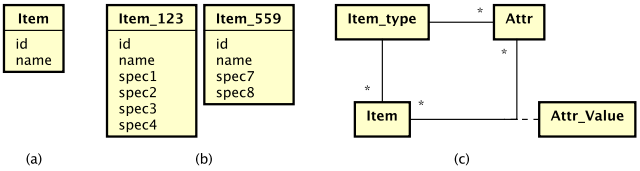

図 5　品目仕様の表現

3.1 属性を外部化する

図 3 の Item クラスでは，品目によって異なる仕様を持てない（図 5a）．では，品目ごとにクラスを設ける方法では，品目が増えたり減ったりするたびにデータベースのテーブル定義やプログラムを変更することになる（図 5b）．今のところの妥当な解決法は，Item クラスの仕様を表現する属性を観測パタン[9]によって外部化すること，すなわち「Item_type（品目型）」「Attr（仕様）」「Attr_Value（仕様値）」の各クラスを設けて図 5c の構造によって Item クラスを修飾することである．これで品目を仕様種と仕様値の組合せによって変数化して(パラメトリックに)扱えるようになる．

3.2 ステレオタイプ<<item>>の導入

このような仕様の外部化を品目ごとに表記するのは煩雑なので，図 6 のようにステレオタイプ<<item>>を定義して，表現上はシンプルに見せる．等号の左側が表記法で，右側が定義である．実装はあくまでも図 5c に基づく．

また，これに合わせて品目間の仕様制約の記述も同様にステレオタイプ<<rule>>を定義して，表現上はシンプルに見せる．この実装は，ドメイン特化言語を設計して，外部関数を記述して集約して格納する方式をとっている．これらのステレオタイプを使って，制約ルールつきのパラメトリックな部品構造を書いたのが図 8 である．

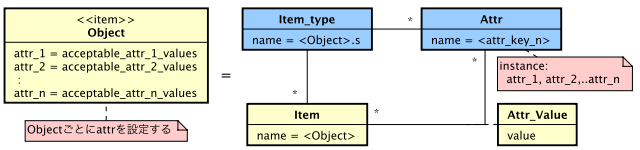

図 6　ステレオタイプ<<item>>の定義

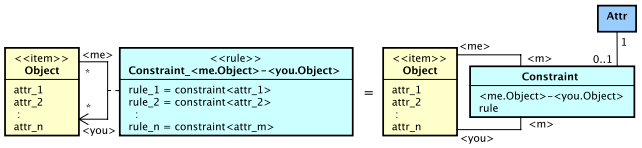

図 7　ステレオタイプ<<rule>>の定義

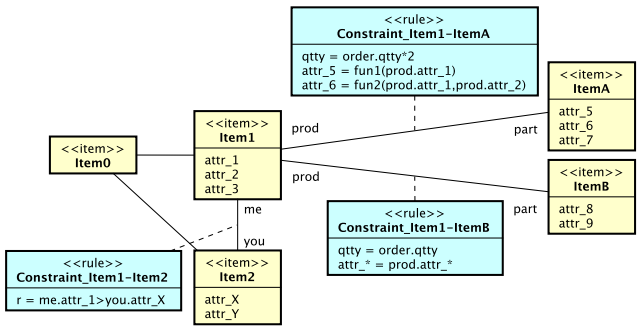

図 8　制約ルールつきパラメトリック部品表の例

この図は，ItemA および ItemB の属性は員数も含めて，Item1 の属性から，Constraint_Item1-ItemA および Constraint_Item1-ItemB で定義される関数で決定されると述べている．また，Constraint_Item1-Item2 は，同じ製品を構成する 2 つの品目（Item1，Item2）間の属性制約を記述している．この記法により，多仕様製品の品目構成をパラメトリックに表記でき，進化し続ける製品の部品表の保守が合理的に扱える．

3.3 「話題沸騰ポット」の部品表を書く

これらのステレオタイプを使って，「話題沸騰ポット」のモデル(クラス図)を書いたのが図 10 である．このモデルは，製品の仕様値に基づいて各部品の仕様値が連鎖的に決定するようになっている．制約記述を 2 つだけ例示した．

図 10 のモデルは図 4 とよく似ているが，図 4 は図 3 左のクラス図に基づくインスタンス図であり，その構造だけが記述されている．図 9 はクラス図でありパラメータである製造注文などに基づいて仕様値を決定することにより，制約ルールの許す範囲でさまざまな品目構成を生み出しうる「知識」であることに注意する．

4. 制御システムのモデル

さて，知識レベルとしての部品表を知った上で，図 1 の検討に戻る．「話題沸騰ポット」の本来の課題は制御システムの分析モデルの設計であった．

4.1 オブジェクト指向分析モデルの対策

「話題沸騰ポット」の機能仕様から新たに設計した制御システムのクラス図が図 11 である．その際に用いたシーケンス図を図 14 に示す．給湯システムは制御システムとは独立とみて別に設計した（図 15）．

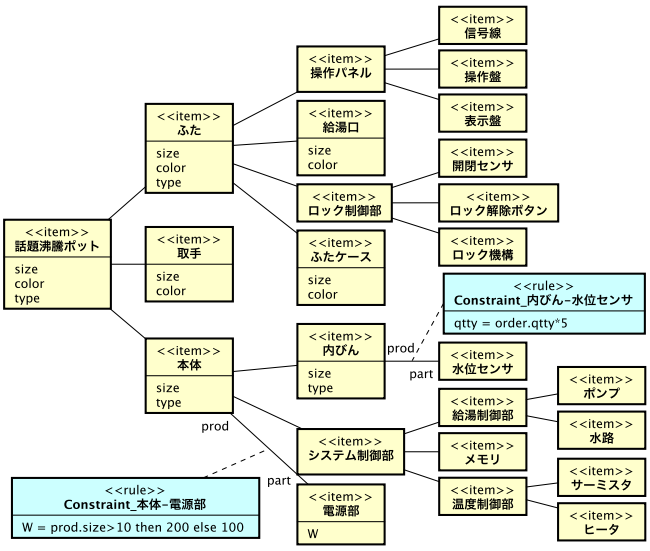

図 9　「話題沸騰ポット」のパラメトリック部品表のモデル（クラス図）

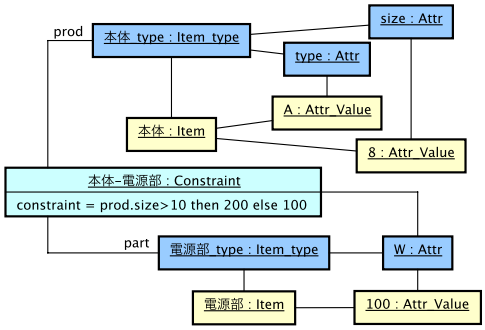

図 10　「話題沸騰ポット」のパラメトリック部品表のモデル（インスタンス図）

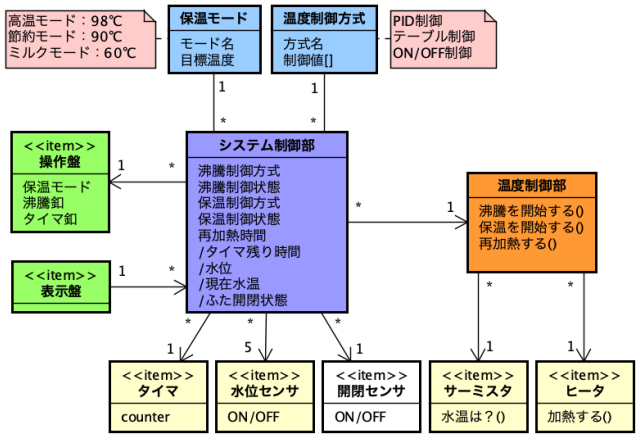

図 11　「話題沸騰ポット」の分析モデルの対案

このクラス図では，上で議論した品目に該当するモノは除いたが，制御システムに関わると判断したものは残した．むしろ，これらは制御システムにおいて interface として定義したい知識である．よって，このモデルには「ふた」クラスも「お湯」クラスも登場しない．なお，このモデル自体も知識レベルにある．

多重度は，「システム制御部」が複数の異なる方式が差し替え可能で，それごとに他のクラスのインスタンスを選択すると想定して設定した．

4.2 組込みソフトウェアとデータ

「話題沸騰ポット」の構造化分析設計版[6]では極めて精緻な分析が行われているが，その中に ER 図で表されるようなデータモデルは示されない．DFD のデータストアとして示されているのは「給湯の状態」や「現在水温」などであり，これらは繰り返され記録されるようなデータではない．これらはむしろ「構造図」と呼ばれるモジュールの呼び出し順を記述した図の中で，末端のセンサが呼び出し時に値を取り込み，その値によって振る舞いを変えるモジュール間の受け渡しデータとして記述されている．

図 11 のオブジェクト指向分析モデルでも，繰り返されるデータは見あたらない．「保温モード」や「温度制御方式」は前もって与えられた制御用データであり，「システム制御部」がもつ属性はセンサ値からの導出データでしかない．

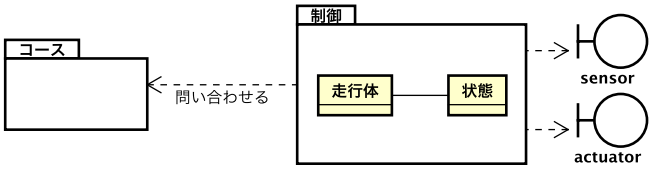

図 12　ET ロボコンにおけるモデリング戦略

4.3 組込みソフトウェアとエンタープライズシステム

本当に，組み込みソフトウェアにはデータが存在しないのだろうか．蓄積されるデータの扱いから見た組込みソフトウェアとエンタープライズシステム[13,14]のモデリングの違いについて考えてみる．

4.3.1 組込ソフトウェアとは

組込みソフトウェアとは，文献[6]によると「ハードウェアと密接に関係しているものです．ハードウェア，もしくはハードウェアを介した事象・状態を読み取り，ハードウェアを制御して利用者に何らかの効果をもたらします」とあり，「ハードウェアの扱いが機器ごとに異なるため，組込ソフトウェアはその機器専用になります」とある．

それゆえ，組込みソフトウェアの特徴は NTCR 条件としてまとめられるという．NTCR 条件とは，自然法則を扱う（Nature），リアルタイム要求が厳しい（Time），制約事項が厳しい（Constrain），高い品質と信頼性（Reliability）にあるとする．

4.3.2 ET ロボコンにおけるモデル

組み込みソフトウェアにはデータはないのかを確認するために，ET ロボコン[15]の競技で用いられたチャンピオンシップ大会まで勝ち残った作品のモデル図を主催者のご好意で提供していただき，これらを参照した．ET ロボコンは，「(略)『組込みシステム』分野における技術教育をテーマに，決められた走行体で指定コースを自律走行する競技」である．

筆者がこの自律走行プログラムをモデリングするならば，マクロには図 12 のような構造を考える．すなわち，コースを記述するオブジェクト群と，それを見ながら走行体を制御するオブジェクト群とに分けた構造である．こうする理由は，コースがどう変わっても問題なく自律走行するように課題をとらえたからである．しかし，実際の作品の分析モデルではコース定義と制御構造が混在しており，単一のインスタンスの組合せによる差し替え不能なそのときだけのモデル（すなわちクラスである必要がない）となっている．これが NTCR 条件のもたらす必然であったとしても，やはり分析モデルは問題をとらえたものであってほしい．

4.3.3 エンタープライズシステムとは

ここでいうエンタープライズシステムとは，企業や組織の中核的業務を執行する情報システムである．これは，組織活動をその目的に沿って最適化するためにリアクティブに働きながら，一方でビジネス戦略やビジネス環境の変化

に合わせてそれ自身変化し，拡張されていく．

エンタープライズシステムでは，繰り返し発生する時刻情報をもったトランザクショナルなデータがシステム内に蓄積され，そのデータがシステムの振る舞いを変化させる．そのデータの意味構造を明らかにすることが分析モデリング作業の中心になる．

4.4 制御システム

「話題沸騰ポット」も利用者にとっていつでも望ましい状態のお湯を得ることが目的であり，エンタープライズシステムも企業のビジネスや組織のミッションを最適化し続けることが目的である．これらはともに環境に対してリアクティブなシステムであり，一種の制御システムと見ることができる．ただし，与えられる環境の範囲が異なり，前者では現時点のデータを観測することでこと足りるのに対して，後者では過去および予定のデータが必要とされ，れ自体が複雑である．そのようなデータを人が観測し，環境に働きかけるという違いがある．

4.4.1 制御システムに見る学習の階型論

組込みソフトウェアもエンタープライズシステムもともに制御システムの形態であるととらえた場合のシステムの特徴を，Engeström[16]の解釈に基づく Bateson の学習の階型論[17]を参考にして評価してみる．

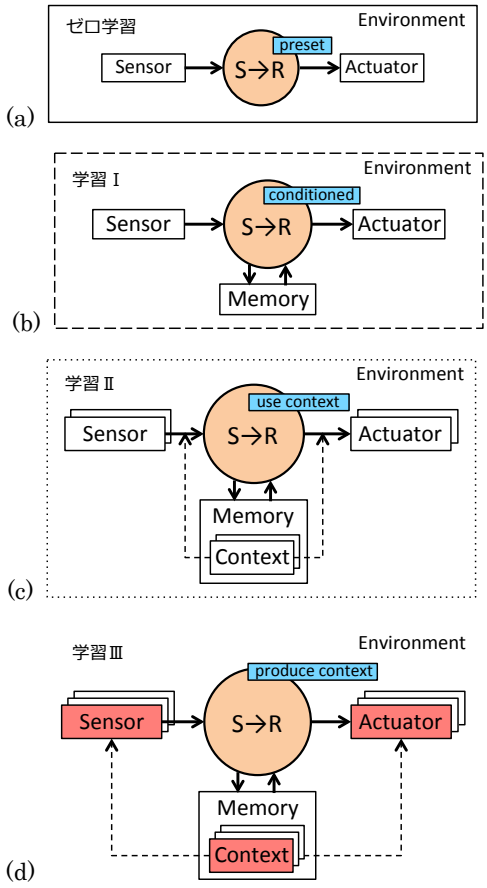

図 13　制御システムを学習の階型論で評価する

4.4.2 ゼロ学習

図 13a は「無条件反射」のシステムである．このシステムは閉環境において，センサを介してあらかじめ決められた刺激（stimulus, S）を得て，あらかじめ決められた規則に従って反応（response, R）に変換し，アクチュエータを介して反応を環境に放出する．フィードバックは環境を介して行われる．このシステムでは現時点のデータのみが扱われるので記憶は必要とされない．「話題沸騰ポット」の制御システムに対応すると考えられる．

4.4.3 学習 I

図 13b は「条件反射」のシステムである．繰り返される刺激と反応の対連合学習による連想記憶に基づき，現在及び過去のデータに基づき定型的な処理を行う．実時間性は必要とされないことが多い．単純な受発注システムなどに対応すると考えられる．

4.4.4 学習 II

図 13c は文脈（context）を獲得したシステムである．刺激と反応の妥当な対連合が文脈によって異なること，文脈が切り替わることを知っている．これが上で述べた「知識」に対応すると考えることもできる．文脈は未来の予想を可能にするので，予定も含めたデータ処理が行われる．たとえば，多仕様製品やサービスの多様化に対応した動的資源引当を伴う受発注システムなどが該当する．環境は比較的開かれており，実時間性が要求されることが多い．多くのシステムは重層しており，システム間での相互作用がある．

4.4.5 学習 III

図 13d は開環境においてシステム自らが新たな文脈と正解を作り出すシステムである．このようなシステムが現存するのかも分からない．そうなると，センサもアクチュエータも現在知られているものではない．

4.5 学習階型によって分析アプローチは異なる

組込みソフトウェアと呼ばれるものは，制御システムの提供形態に過ぎないことがわかる．技術的に可能であれば，図 13d のようなシステムを組込みソフトウェアとして提供してもいいからである．

それよりも，学習の階型によって用いる分析モデリング手法が異なることが示唆された．例題の「話題沸騰ポット」のような無条件反射のシステムで，観測時点のセンシングデータにしか関心がないのならば，クラスの概念を導入する意味はない．機能分割の原理に基づく構造化分析が適当と思われる．将来，「話題沸騰ポット」購入した家庭ごとに刺激反応変換の規則を動的に入れ替えられるようになるb のなら，過去データの蓄積と意味処理のための機能が必要となり，集合論を背景理論とするデータ中心の構造化分析やオブジェクト指向分析の意味が出てくる．

文脈の獲得やその変化を扱うシステムの分析には，おそ

b 音声入力は sensor の形態が変わるだけで，変換規則は変わらない．

らく，データベースの正規化手続きに基づくデータモデリングではなく，第 3 章で述べたような知識レベルあるいはべき集合に基づくモデリング手法の確立が必須であろう．ステレオタイプは UML の拡張機構であって，それが本質的問題解決の方法ではない．

文脈を自ら生成するシステムに対する分析アプローチはまだ分からない．

5. 今後の課題

ここまで,「話題沸騰ポット」の部品構造と制御システムの構造とは，関心が異なるとして分離して論じてきた．しかし，分離したものは最終的にシステムとしてつなぎ直されなければならない.「ふた」をクラスにしてしまった理由もここにあるのかも知れない．

たとえば，制御システムのモデル（図 11）にある「表示盤」は部品構造のモデルにある（図 10）にある「表示盤」とどう対応づくのかは明示されていない．同じ名前にするべしという「お約束」に頼るのではなく，何らかのモデルにおいて明確な設計意図（アーキテクチャ）として記述されなければならない．

これは，組込みソフトウェアに特有の課題ではない．エンタープライズシステムでも同様である．たとえば，分析されたクラスがどのような実装クラスになり，それがどのように実行可能ファイルとして媒体上に存在し，それがどのような基盤のサーバにロードされ，さまざまな基盤のサービスとどのように連携して稼働するのかを記述する手段はない（部分的には試行しているが）．時間的に後で決まる設計要素が，前で決めた設計要素に遡って反映されることは稀であり，それがゆえに前段階の関心事から除外されてしまうからである．ましてやクラウド上で動的に複製されたり消去されたりする状況をどのように追跡すればよいのだろうか．DevOps と呼ばれる開発チームと運用チームのコミュニケーションの分断の本質的な原因の一つはここにあるのかも知れない．

通信ネットワーキングにおいても同様の課題があると聞く．ドメイン層のモデルとプレゼンテーション層のモデルとの対応関係についても同様である．以上については今後の課題である．

6. まとめ

「話題沸騰ポット」のモデルに対する疑問から出発して，モデリングの関心を分離した上で，現実世界のモノをそのままオブジェクトとして取り上げるべきでないと述べた．

また,あらゆる情報システムを制御システムと見立てて，Bateson の学習の階型論に基づいて制御システムの動作原理をカテゴライズし，分析モデリング手法の妥当性を検討した．今のところは，ゼロ学習レベルの制御システムの分析モデリングでは，データの集合が想定できないため，クラスに基づくオブジェクト指向分析を適用するのは妥当でなく，構造化分析の機能分解が有効であること．学習Ⅰレベルの分析モデリングでは，バリエーションを持ったデータの集合が想定できるため，クラスに基づくオブジェクト指向分析を適用するのは妥当であること．学習Ⅱレベルの分析モデリングでは，知識レベルを扱える分析モデリング手法が必要であること．そのための UML の拡張を提案したが，これで十分とは言えない．学習Ⅲレベルに対するモデリング手法はわかっていない．なお，以上は分析モデリングに限定され，設計および実装のためのモデリングには該当しない．

さらに今後の課題を挙げた．すなわち，設計から実装，運用に至る設計意図連鎖としてのトレーサビリティの分断問題である．これについても広範な研究が進むことを期待する．

謝辞

SESSAME WG2 のメンバの皆さんに感謝します．皆さんの研究成果を基に,多くのことを学ぶことができました．ET ロボコン主催者である一般社団法人組込システム技術協会様に感謝します．組込ソフトウェアにおける分析モデリングの実情を知ることができました．原稿を下読みしてくださった方々に感謝します．多くの改善点が見つかりました．

参考文献

[1]磯田定宏,「実世界モデル化有害論-オブジェクト指向モデル化技法の解明」,電子情報通信学会論文誌. D-1, 情報・システム 1-情報処理 Vol.83(9), pp.946-959, 2000-09-25.
[2]平澤 章,「オブジェクト指向でなぜつくるのか」,日経 BP, 2004
[3]Meyer, B.: *Object-Oriented Software Construction*, Prentice-Hall, 1988. 邦訳)二木厚吉監訳，酒匂寛，酒匂順子訳：「オブジェクト指向入門」，アスキー出版，1990.
[4]SESSAME WG2,「組込ソフトウェア開発のためのオブジェクト指向モデリング」，翔泳社，2006
[5]松澤芳昭，児玉公信，塩見彰睦，酒井三四郎，「PCA と MVC の複合アーキテクチャスタイルを用いた組込みモデリング技法の提案」，組込みシステムシンポジウム 2011 論文集, pp. 17-1-17-8, 2011-10-12.
[6]SESSAME WG2,「組込ソフトウェア開発のための構造化モデリング」，翔泳社，2006
[7]Wilson, B.著，根来監訳,「システム仕様の分析学−ソフトシステム方法論」，共立出版, 1996.
[8]Stevens, P. & Pooley, R.: *Using UML: Software Engineering with Objects and Components*, Pearson Edu. 1999. 邦訳)児玉公信監訳：「オブジェクト指向とコンポーネントによるソフトウェア工学」，ピアソン・エデュケーション，2000.
[9]Odell, J. J., *Advanced Object-Oriented Analysis & Design Using UML*, SIGS Referenced Library, Cambridge Univ. Press, 1998.
[10]Fowler, M.: *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1997. 邦訳)児玉公信ほか訳,「アナリシスパターン」，ピアソン・エデュケーション, 1998.
[11]児玉公信，水野忠則,「少量多品種型生産管理システムの一般モデル CHARM の提案」，情報処理学会論文誌, Vol. 49(2), pp.

902- 909, 2008.
[12]Opitz, H.: *Group Technology,* 196X. 邦訳)鈴木 隆ほか訳,「グループ・テクノロジー―部品グループ加工によるコストダウン」, 日本能率協会, 1969.
[13]児玉公信,「計画・実行システムの一般モデル―生産管理システムと金融業務システムの共通性―」, 情報処理学会研究会報告, IS109-4, 2009-9-14.
[14]児玉公信,「取引処理のための基幹情報システムの状態側面から見た一般モデル」, 情報処理学会研究会報告, IS141-4, 2017-8-25.
[15]http://www.etrobo.jp/2019/ (2019-02-05 accessed)
[16]Engeström, Y., 山住勝広ほか訳,「拡張による学習―活動理論からのアプローチ」, 新曜社, 1999.
[17]Bateson, G.: "The Logical Categories of Learning and Communication," 1972. 邦訳)佐藤良明訳,「学習とコミュニケーションの階型論」, in 精神の生態学, 新思索社, pp.382-419, 2000.

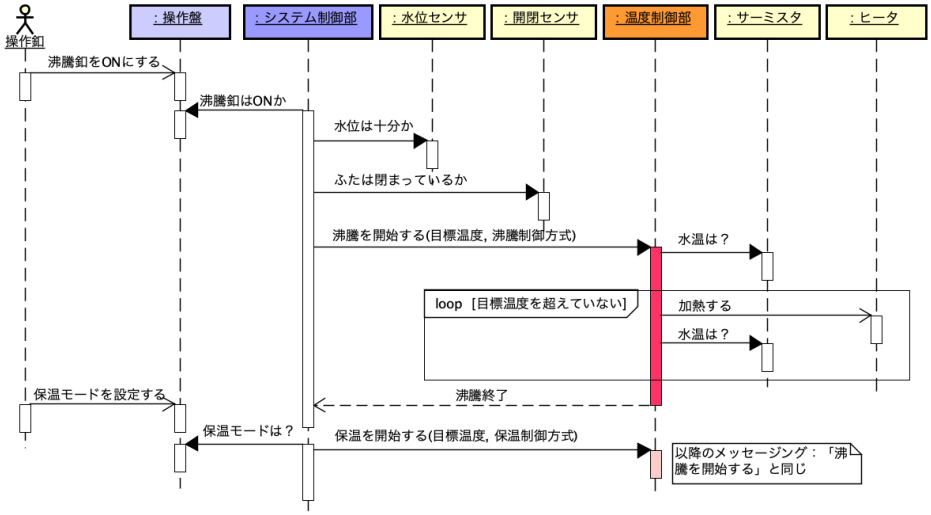

図 14 「話題沸騰ポット」の分析モデルのためのシーケンス図(一部)

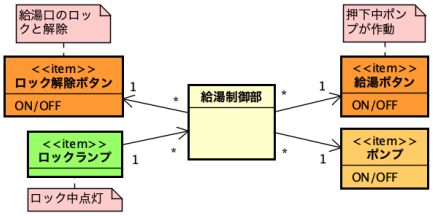

図 15 「話題沸騰ポット」の給湯システムの分析モデル