数理解析研究所講究録
591巻 1986年 175-182

ソフトウェアモデリングと対象モデルの変換に関する一考察

富士通㈱国際情報社会科学研究所　西田　泰伸 (Taishin Nishida)
小林　要　(Kaname Kobayashi)

1．はじめに

これまでもソフトウェア技術者は，たとえ明確に意識していないにしても，ソフトウェアの設計・開発・保守においてモデル化を行ってきたはずである。それらの，ソフトウェア・ライフサイクルに現れるモデルの役割を示したのが図1である。ここで，ソフトウェアが利用者の手にわたって意味のある働きをする現実の世界を，そのソフトウェアの対象世界という。ソフトウェアの設計者は，対象世界の知識をもとに問題を解決し自分なりの解決のイメージをつくる。また，ここでいう中間記述は，形式的仕様記述やコンパイラ言語によるソースプログラムを想定している。これらは，究極的に計算機で実行されるから，抽象的計算論の世界にその意味のモデルがある。

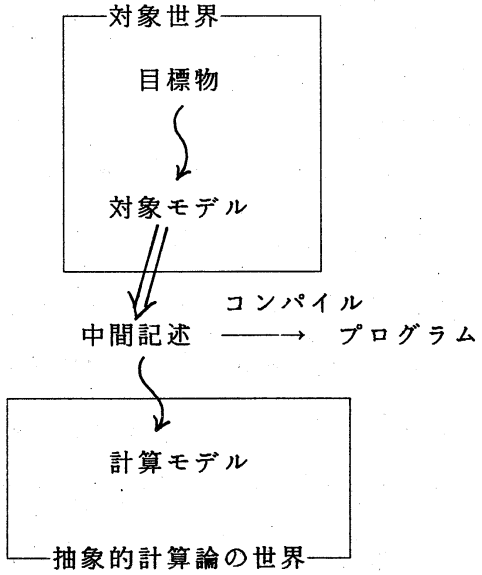

図1

図1に現れる「自分なりの解決のイメージ→フォーマルな中間記述」の段階は，様々な用語で取り上げられている，ソフトウェア工学の中心的概念である［5］。たとえば，「要求定義」，「仕様記述」，「××設計」，「プロトタイプ」などは皆，図1の⇨の部分を指向していると考えられる。

しかしながら，上にのべた従来の考え方では，あくまで「計算機にのせる」ことに主眼があったので，「「計算機の都合に合せた」フォーマルな中間記述の都合に合せて」解決のイメージを作る，というきらいがあった。本論文では，計算機より対象世界の都合にあわせたモデリングによる，ソフトウェア開発の支援の可能性について二三の考察と提案を行う。

2．対象モデル

図1で対象世界とフォーマルに定義された記述との関係を抜き出すと，

S_0，S_1，S_2，．．．：対象世界における解決

↑　↓　変換・対応

F_0，F_1，F_2，．．．：FORMALに定義された記述・解

となる。S_0，S_1，．．．は設計者の頭の中にある具体例(sample)あるいは解決(solution)である。これらは，思考途上にある明確化されていない事項，あるいは暗黙に仮定されている情報を多数含んだ，断片的"Thinking matter" の集まりである。それらに対応して，実際に問題を解くためには，必要な明確化を行い暗黙の仮定をおぎなって定式化したF_0，F_1，．．．を作る必要がある。

ここで，S_0，S_1，．．．といった解決のイメージを，対象世界の都合を重視してフォーマルなものへ変換・対応付けすることが，対象世界のモデリングである。これは，その対象世界に合せた問題解決のフレームワークを与えることにほかならない。その際に，対象世界とフォーマルなシステムの組合せに応じて，両者の変換・対応の性格が決る。そうして，対象世界に応じた適切なフォーマルなモデルの記述が定まれば，それを計算機にシステム化して，ここで述べた設計者の段階的な設計過程の支援が可能になる。

この対象モデリングの考え方がよく現れている例として，3節で金融機関の日常の業務上の用語による業務手続からＣＯＢＯＬプログラムへの自動変換を考察する。4節では，プロトコルの設計においてＦＳＭによるモデル記述の改良に関する一方式を提案する。

3．金融業務からＣＯＢＯＬへの変換

金融機関の内部的な業務用語によって規定された業務手続から，ＣＯＢＯＬプログラムへの自動変換システムが報告されている（木村［３］）。このシステムでは，日本語の業務用語による業務手続の記述を日本語専用プログラムと呼び，入力の日本語専用プログラムを予め与えておいた変換規則とパターンマッチさせてＣＯＢＯＬプログラムを出力する。このシステムにより，開発効率が２～３倍に向上したという。変換されるキーワードと日本語専用プログラムの記述例を表１に示す。

基本業務動作	ＣＯＢＯＬキーワード	業務手続記述例（日本語専用プログラム記述例）
取り出す	ＯＰＥＮ	無通帳データファイルを入力用にオープンする
読む	ＲＥＡＤ	無通帳データファイルを読んで，データの読込みが終了したら，１を終了判定に記入し，終了する
判定	ＩＦ	顧客変り処理のため，口座番号が口座番号控に等しくない場合には，見出し処理を行う （「ため」以前が注釈）
目的	（注釈）	
記入・転記	ＭＯＶＥ	口座番号を印刷のため，口座番号を口座番号欄に記入する
計算	COMPUTE	利息　＝　残高積数　×　利率　÷　３６５
書く・印刷	ＷＲＩＴＥ	無通帳取引明細表を印刷し，明細表の頁を１行行送りする
行う	ＧＯ　ＴＯ	データが終了するまで，プリントファイル編集処理を行う
保管	ＣＬＯＳＥ	無通帳データファイルをクローズする

表１（木村［３］より改変）

このシステムを対象モデリングとしてみた時に，次の点で日本語ＣＯＢＯＬと異なる。

ｉ．金融機関の業務を優先して入力の構文を決めた。

ii．変換にあたってどの変換規則ともマッチしない部分がある時は，その都度新らたな規則を付け加える。

ｉについては，金融手続記述からＣＯＢＯＬプログラムを作成する場合には，プログラムにexplicitに書かなければならないことがらが，業務動作にすでにあるか，あるいは直

接的に対応付け可能である。すなわち，金融業務の対象世界とＣＯＢＯＬというフォーマルな記述の組合せを選んだことで，巧まずして対象世界優先のモデリングになった。そのために比較的単純な日本語化の手法でも，これだけの効率の向上が得られたのだと考えられる。iiに関しては，設計者の思考段階が進むにしたがって，これまで明確化されていなかった情報を新たに変換システムに取り込むことができるので，効率の大幅な向上が達成されたと考えられる。

４．通信プロトコルのモデル＊

有限状態機械（ＦＳＭ）を用いたモデルによるプロトコルの設計・検証の自動化の研究が行われている［１，２］。このモデル化では，プロトコルに現れる二つの端局をＦＳＭで現し，それらの間を先入れ先出し（ＦＩＦＯ）型のバッファで連結する。しかし，これまでの研究では，異常動作を検証の対象にする時は，調べなければならない状態の数が非現実的に増大してしまった。これは，プロトコルの設計においては，対象世界の動作すべてをexplicitにモデル化する作業における”セマンティク・ギャップ”が，金融業務などと比べるとたいへん大きいことを意味する。

ここでは，仕様記述から決定できる正常動作と，それ以外の異常動作の独立性に注目し，プロトコル設計を階層化したモデルを提案する。そのなかで，異常動作の階層にエラー処理を依頼するために，異常があった時の状態を特定する一方式を考察する。

４．１　プロトコル設計の階層

プロトコルに求められるすべての機能を同一に考えるのではなく，「ユーザの通信要求の処理」，「通信サービス」，「トラブルの処理」など役割に応じて階層化する（図２）。ここで，各階層は外から見た時はインタフェースのみによって規定され，内部は他の階層から影響を受けない。

＊本節は準備の都合で講演発表には含まれなかったが，本研究の重要な部分であるのでここに入れた。

これまでのＦＳＭを用いたモデルは「通信サービス」層を規定していた。「ユーザの要求」階層では，どの資源を使うかなど，ユーザが直接指定しない事柄を明確化して通信サービスのパケットを作り，「通信サービス層」に渡す。「トラブルの解決」層では，「通信サービス」層で発生するトラブルを解析して，解決に必要な情報を補い，本来のユーザの要求を満足させるように解決する。このように，独立した階層をもうけることにより，各階層で局所的に正当性を保証しておけば全体の正しさも満たされることになる。

ここではＦＳＭで表された「通信サービス」層で発生するトラブルを解析するために，トラブル発生時に状態を特定する一つの方式をのべる。ここで提案する方式では，解析すべき状態を，設計者の意図していた状態に限定することができる。

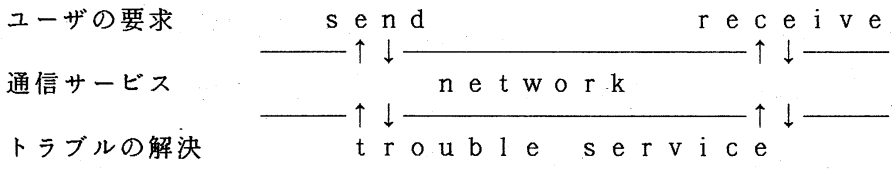

図２　設計の階層

４．２　通信プロトコルの有限状態モデル

二つのＦＳＭ（Ｍ，Ｎ）がＦＩＦＯ型のバッファによって互いにむすばれたネットワーク［Ｍ，Ｎ］で，通信プロトコルがモデル化できる［２］。ここで，Ｍ，Ｎはラベル付（有限）有向グラフであり，出力にあたるsending edgeには－ｇ，入力にあたるreceiving edgeには＋ｇのラベルをつける。ただしｇはメッセージとする。v(w)がＭ（Ｎ）のノード，ｘおよびｙがメッセージの系列の時，四字組(v,w,x,y) を，ネットワーク［Ｍ，Ｎ］のGlobal Stateと定義する。

二つのglobal state(v,w,x,y)，(v',w',x',y')が次の四つの条件のどれかを満たす時，(v,w,x,y) から(v',w',x',y') へ遷移するといい，(v,w,x,y) → (v',w',x',y')と書く。

ｉラベル－ｇの付いたsending edge v→v'がＭにあり，w'＝w，x' ＝x，y' ＝yg

ⅱラベル－ｇの付いたsending edge w→w'がＮにあり，v'＝v，x' ＝xg，y'＝y

ⅲラベル＋ｇの付いたreceiving edge v→v'がＭにあり，w'＝w，gx'＝x，y' ＝y

ⅳラベル＋ｇの付いたreceiving edge w→w'がＮにあり，v'＝v，x' ＝x，gy'＝y

初期状態（v_0, w_0, ε, ε）（εは空語）が存在する。ただし，v_0, w_0はそれぞれM，Nにおける特別のノードである。

4．3　トラブルの解析

ここでは，トラブルとして通信路のエラーを考え，そのなかでもメッセージのロスに限定する。良いエラー検出コードが開発されており，エラーが検出された時はそのメッセージを捨てるのが一番簡単な対応策だから，メッセージのロスは，おこり得る通信路のエラーのうちでも最も一般的なものだと考えられる。

M，NをＦＳＭとし，Mにおいてはラベル－ｇのついたsending edge，Nにおいてはメッセージｇを受信するラベル＋ｇのついたreceiving edgeがあるとする（図３ａ）。ここでメッセージｇが失われたとすると，Mでは，相手になんの影響も与えずに自発的にその状態が変化するから，sending edgeがオートマトン理論で言うε遷移になり，Nにおいてはメッセージが来ないかぎり状態は変化できないから，ループすると考える（図３ｂ）。メッセージロスがあった時の，ＦＳＭの論理的に等価な動作をこれによって定義する。

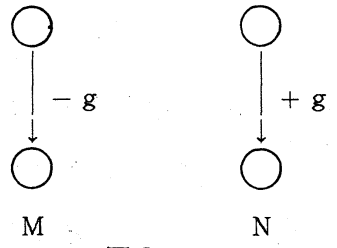

図３ａ

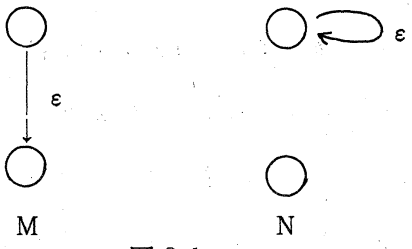

図３ｂ

われわれの目的は，メッセージロスがglobal stateの遷移に及ぼす影響を解析し，それを「トラブルの解決」へのインタフェースとすることである。つぎの性質はそのために有用である。

性質１　Global stateの遷移において次の図式が成立する時（未定義受信がないかぎり必ず成立する），

$$\begin{array}{ccc} (p_0,\ q_0,\ x,\ y) & \rightarrow & (p_1,\ q_0,\ x,\ yg) \\ \downarrow & & \downarrow \\ (p_0,\ q_1,\ x,\ \varepsilon) & \rightarrow & (p_1,\ q_1,\ x,\ g) \end{array}$$

メッセージgが失われた状態遷移においても次の図式が成立する。

$$\begin{array}{ccc} (p_0,\ q_0,\ x,\ y) & \rightarrow & (p_1,\ q_0,\ x,\ y) \\ \downarrow & & \downarrow \\ (p_0,\ q_1,\ x,\ \varepsilon) & \rightarrow & (p_1,\ q_1,\ x,\ \varepsilon) \end{array} \quad \square$$

すなわち，Mがメッセージgをバッファに送り出すのと，Nがすでにあるバッファのメッセージ系列を受信するのが交換可能な時，メッセージgが失われた状態遷移もNによる受信と交換可能になる。これはM，Nの動作を別々に追跡すればすぐ確かめられる。

ひとつのメッセージロスがあった時のglobal state遷移に及ぼす影響は次の手順で解析できる。

手順 i．エラーのない時におこり得るすべてのglobal state遷移を求める。

ii．受信側のバッファが空になっているstate からの送信状態遷移をメッセージロスによる遷移に変える。

iii．iiで得られた新しい状態から到達するすべての状態を求め，デッドロックや未定義受信など設計者が意図しない状態・動作がないか調べる。

手順 i では，エラーがないと想定した時に設計者が意図したglobal state遷移（たとえばfair progress ［2］）だけを考えればよい。さらに，性質1により，手順ii では受信側のバッファが空になっているstate だけ考慮すればよい。これによって考察すべき状態数は相当減らせる。

5．おわりに

この論文では，ソフトウェアの設計のために，「対象世界の都合」を重視したモデリングの考え方を論じてきた。設計段階でモデリングを十分行う考えは，モデルを完成したソフトウェアの模型（実現モデル）だとみなせば，ラピッドプロトタイプになろう。ここでは，できあがったソフトウェアとは別の形態で，対象世界ごとにモデルがあるとして，実例の考察を行った。このような対象世界ごとのモデルを，計算機にシステム化すれば，そ

の対象世界のためのソフトウェアＣＡＤができる。

対象世界のモデリングにおいて，モデルの外にある知識とモデリングシステムのインタフェイスが重要になる。３節の金融手続の変換では，これまでの規則では変換できない業務手続を変換するための知識，４節のプロトコルのモデルでは，「ユーザの要求」を「通信サービス」や「トラブルの解決」に反映させるための自然言語解析の知識，を取り込む必要がある。これらの知識をどのように表現するかが対象モデリングの今後の課題である。

文献

[１] Bochmann, G. V.: Finite State Description of Communication Protocols, Computer Networks 2 (1978), 361-372

[２] Gouda, M. G. and Han, J.: Protocol Validation by Fair Progress State Exploration, Computer Networks 9 (1985), 353-361

[３] 木村　俊一：口語体による日本語プログラムの実現とその評価，情報処理学会全国大会，１９８６年３月.

[４] ソフトウェアモデリング夏季シンポジウム報告書，富士通㈱国際情報社会科学研究所，１９８５.