プロブレムフレームに基づく組込みシステムの状態遷移分析支援システム

紫合 治[†]　　横山 薫[††]

本論文では，ソフトウェア開発の初期段階において有効であるプロブレムフレームに基づく要求分析支援システムについて述べる．システムは要求分析の工程に沿って，プロブレム図，ドメイン状態遷移図，要求状態遷移図の作成を支援し，それらの間の妥当性をチェックした後，マシンの仕様となる状態遷移記述を自動生成する．このとき，プロブレム図で定義したイベントや状態をそのままドメイン状態遷移図で活用し，ドメイン状態遷移図で定義した状態をそのまま要求状態遷移図で利用するなどにより，誤りのない要求分析を支援する．さらに，これらの状態遷移図間の整合性を調べて，要求の漏れや誤り等を自動的にチェックする機能をもつ．

State Transition Analysis System for Embedded Systems by Problem Frames

Osamu Shigo[†]　and　Kaoru Yokoyama[††]

This paper describes a total support system for state transition diagrams appeared in the problem frames. The system provides graphical editors for problem diagrams, domain state transition diagrams as domain properties and requirement state transition diagram, all of which are related together. By analyzing these diagrams, the system automatically generates the machine state transition description. In the system, event names defined in the problem diagram are appeared in the related domain state transition diagrams and domain state names defined in the domain state transition diagrams are appeared in the requirement state transition diagram to describe the required domain states. This significantly reduces the effort to draw new diagrams. Also, by checking the consistency among these diagrams, the system reports the requirement errors, including the leakage of requirements.

1. はじめに

ソフトウェア開発の分析・設計では，UML[1]やその支援ツール[2][3]が広く使われている．上流工程である要求分析においても，ユースケース図，シーケンス図，アクティビティ図，状態遷移図，クラス図などが活用されている．しかしながら，UML は元来ソフトウェアの構成や振舞いを設計するためのものであり，いわば問題の解決策を設計するためのツールといえる．

一方，問題の解決を考える前に問題そのものに集中し，問題が存在する現実世界を分析・構造化し，さらにパターン化していく手法として，Jackson のプロブレムフレーム[4]の考え方が注目されている．システムの開発の初期段階においては，システムそのものを考える前に，システムを取り巻く現実世界を明確に把握することは重要であり，プロブレムフレームの活用の効果が期待できる．

UML を活用する場合，図形描画だけでなく，図のエラーのチェックや関連図との関連のチェックなども含めて，ツールの活用が有効である．プロブレムフレームに対しても，プロブレム図の描画だけでなく，そこに現れるドメインのプロパティ定義や要求記述のための状態遷移図等，各種の図形の描画，図形間の関連のチェックや関連情報の生成等の機能を持つツールによって，分析作業の効率化が期待できる．

本論文では，プロブレムフレームの中で組込みシステムによく出てくる「必要な振舞いフレーム」に限定した支援システムについて述べる．フレームを限定することによって，支援システムの適用範囲は制限されるが，より高度な支援機能を持つことが出来る．プロブレムフレームによる組込みシステムの状態遷移設計手法として，著者等はジャクソン法による状態遷移設計手法[5]を提案したが,その手法を効率よく適用するには支援ツールが必須であると考えた．ここで述べるシステムは，プロブレム図をベースにドメインや要求の状態遷移図を描画し，マシンの状態遷移記述の自動生成までを含んだ統合支援システムで，上の手法の支援ツールとして使うこともできる．

2. プロブレムフレームと状態マシン

Jackson はさまざまな問題を５つの型(Problem Frames)に分類し，それぞれについて考察している[4]．ここでは，このうち，組込みシステムによく現れる振舞い問題(必要な振舞いフレーム)に限定して考察する．

† 東京電機大学　情報環境学部
School of Information Environment, Tokyo Denki University
†† ＮＥＣソフト株式会社
NEC Soft, Ltd

2.1 プロブレム図

図１にプロブレムフレームで基本となるプロブレム図の構成を示す．図で，２本の縦線をもつ要素をマシンと呼び，問題の解決策を担当する．マシンと関連する四角の要素を*ドメイン*と呼び，外部環境となる部品や装置を表す．また，破線の楕円で，これらのドメインに対する*要求*を表す．

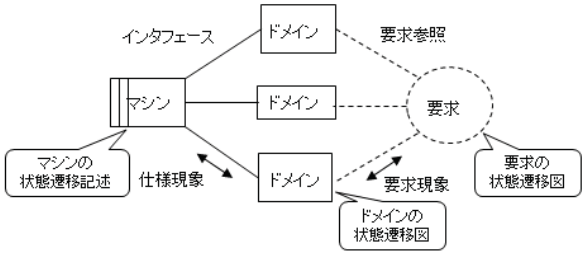

図１　プロブレム図

マシンとドメインをつなぐ実線を*インタフェース*，要求とドメインをつなぐ破線を*要求参照*と呼ぶ．なお，要求とマシンは直接的な関連を持たない．つまり，要求は解決策に対しては何も要請しない．インタフェースや要求参照にそって，線でつながれた要素間で*共有する現象*を規定する．インタフェースと要求参照の共有現象をそれぞれ*仕様現象*，*要求現象*と呼ぶ．仕様現象は，マシンからドメインへの装置の動作指示である(マシンの)出力現象と，ドメインからマシンへのセンサの結果等の(マシンの)入力現象を含む．一方，要求現象は，装置がどうなっているかの状態を示すことが多い．表１に，Jackson の本[4]に現れるいくつかのドメインの例について，仕様現象と要求現象を示す．表では，仕様現象はイベント，要求現象は状態となっている（要求が強制する現象が状態）．ここでは，仕様現象はイベント，要求現象は状態に限定する．

表１　仕様現象と要求現象

問題	ドメインの例	仕様現象	要求現象
片側交互通行用信号	信号機	赤と緑のランプへのパルス	進め状態，止れ状態
走行距離計表示	走行車	車の回転毎に出るパルス	スピードと走行距離
周期と範囲の入力	周期＆範囲	編集機能	データ値
炉の操作	炉設備	炉の操作（点火，送風等）	燃焼中，停止等の状態
水門制御	門扉＆モータ	逆（順）回転，On/Off，位置	閉鎖状態，開門状態等

2.2 状態マシン

振舞いフレームの場合，ドメインは，仕様現象(イベント)と要求現象(状態)の関連を状態マシンとして規定する．また，マシンの仕様として，関連するドメインのイベント制御を状態マシンで規定する[4]．一方，要求は必ずしも状態マシンで記述するとは限らないが，Jackson の片側交互通行用信号や水門制御などの例でも見られるとおり，ドメインの状態変化の要求を状態マシンで表すことも多い．著者等は，ドメインの状態マシンの対応関係の規定からマシンの仕様となる状態マシンを生成する手法を提案したが[5],そこでの対応関係の規定は要求を状態マシンで表した形式になっている．ここでは要求も状態マシンで記述するという制限を設ける．但し，不完全な要求への対応として，断片的な状態マシンも扱えるようにした．この場合，ドメインの状態マシンの規定により，システムが遷移の漏れ等につきエラー表示するので，それらを修正していくことで完全な状態マシンを得ることが出来る(4.2 参照)．

ドメインの状態マシンは，そのドメインがどうなっているかを規定したもので，所与の記述であり，問題の分析において，考え出すものではなく，事実として与えられるものである[4]．これは，設計対象マシンに対するポートのプロトコルを規定した Interface Automata[6]と同様の記述とする．要求の状態マシンは，ドメイン達がどのようになるべきかどのように振る舞うべきかを，ドメインの状態の組合せによって規定したもので，願望の記述[4]となる．これは，状態の中に世界の状況規定をドメインの状態の組として含む Kripke 構造[7]や,状態指向の状態マシン[8]と考えることが出来る．マシンの状態マシンは，ドメインからの入力イベントに反応して，どのように出力イベントを出していくかを規定したもので，UML の振舞い状態マシン[1]に対応する．これらを使って，システムによる要求分析支援の流れは以下のようになる．

① プロブレム図を作成し，共有現象(イベントと状態)を規定．
② ドメイン毎に，①のイベントと状態の関係をドメイン状態マシンで規定．
③ ドメイン達の状態の振舞いの条件を要求の状態マシンで規定．
④ ①～③から，マシンの状態マシン(仕様)を自動生成．

3. 分析支援システム

前節で述べたプロブレム図と各種状態マシンを作成し分析するためのシステムを開発した．このシステムは，問題全体の枠組みの規定としてプロブレム図を作成し，そこのドメインについて，ドメイン状態遷移図を，また，要求に対して要求状態遷移図を作成する．さらに，マシンに対しては，仕様となる状態遷移記述を自動生成することが出来る．ここで説明するドメインの状態遷移図，要求の状態遷移図，マシンの状態遷移記述についての形式的な規定については，文献[5]を参照されたい．

3.1 プロブレム図作成

図 2 はプロブレム図作成ツールによる電話交換システムのプロブレム図の作成例を示す．図の画面中，右上部分に要素や接続線の種類を示すラジオボタンが，右下部分に，インタフェースや要求参照の共有現象を示す，共有現象（イベント・状態）管理表がある．

プロブレム図の入力は，最初にノードとなるマシン，ドメイン，要求を作成し，次に，２つのノードを指定してアークとなるインタフェースや要求参照を引く．ツールはプロブレム図のルールに外れた図の描画に対してはエラーを出すようにしている．例えば，1 つのプロブレム図にはマシンと要求はそれぞれ 1 つしか作成できないよう，２つ目の描画ではエラーメッセージを出すようにしている．

図右下の共有現象管理表には，インタフェースの仕様現象である入出力イベントや，要求参照の要求現象であるドメインの状態を規定する．インタフェースに対しては，管理表の from がドメイン名の行には入力イベントを，マシン名の行には出力イベントを記述する．要求参照に対しては，from がドメイン名の行にはそのドメインの状態名を記述し，from が要求名の行には何も記述しない．

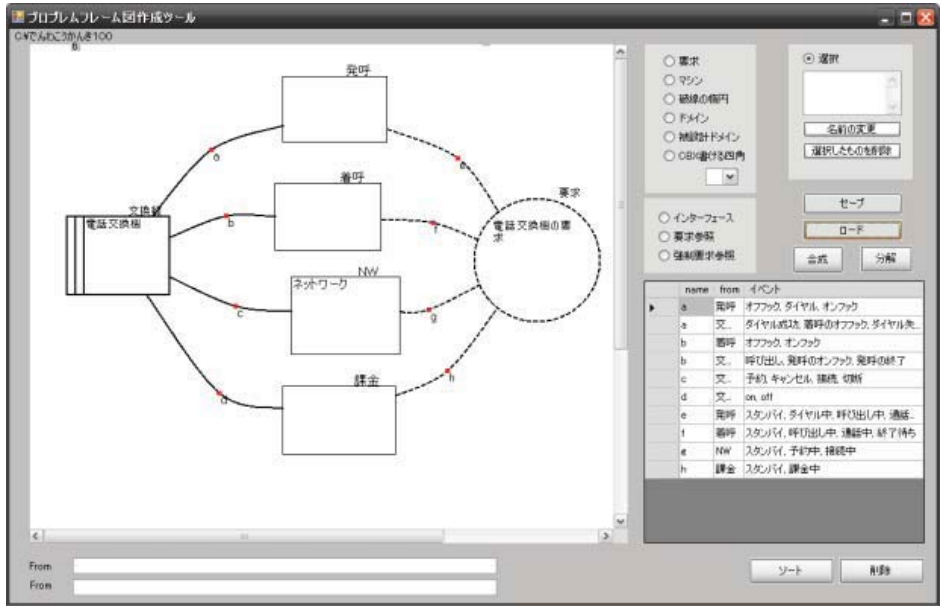

図 2　プロブレム図の作成画面

3.2 ドメインの状態遷移図作成

ドメインの状態遷移図は，各ドメインがどのようなイベントによってその状態がどのように変化するかを表した図である．ドメインの状態遷移図は，プロブレム図と互換性を持ちながら作成する事が出来る．図 3 はドメインの状態遷移図作成ツールを用いて作成した，電話交換機問題における発呼ドメインの状態遷移図である．ドメインの状態遷移図での遷移は，１つの入力または出力イベントを伴う．

ドメインの状態遷移図作成ツール画面を開くには，プロブレム図において作成したいドメインをダブルクリックする．黒丸が開始点，角の丸い長方形は状態を表し，矢印と入出力イベントの記述によって遷移を表す．入力イベントと出力イベントは”／”を用いた表記による区別を行い，出力イベントの前に”／”を記述する．ドメインの状態遷移図はプロブレム図の共有現象管理表と互換性を持ち，状態を登録するときにはそのドメインの状態名のリストが表示されるので，そこから選択することによって状態名を指定できる．同様に，遷移の登録では，そのドメインの入出力イベントのリストが表示され，そこから選択することでイベントを指定できる．指定したい名前が選択リストになかった場合は，状態名やイベントをその場で記述することができ，この場合は逆にここで記述した情報がプロブレム図の共有現象管理表に反映される．

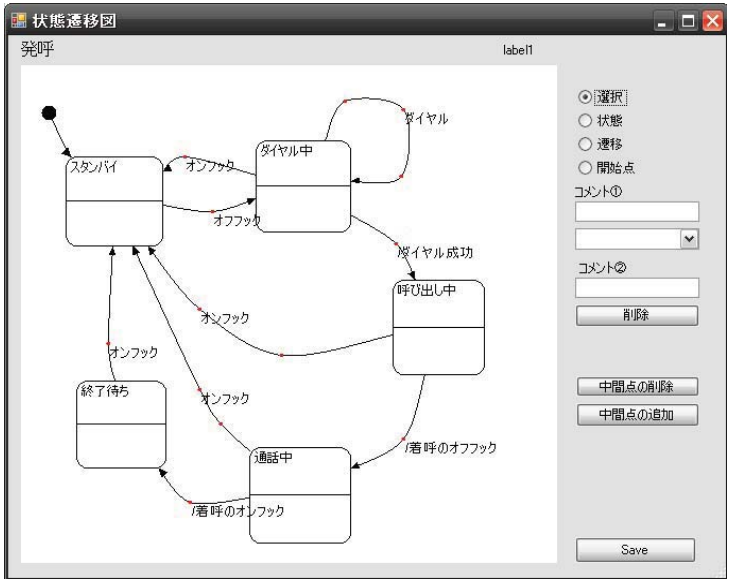

図 3　発呼ドメインの状態遷移図

3.3 要求の状態遷移図作成

プロブレム図を作成し，その全てのドメインに対してドメインの状態遷移図を完成する事により，要求の状態遷移図の作成を開始する事が出来る．要求の状態遷移図では，タイムアウトか，ドメインからの入力イベントに対して，全てのドメインの状態がどのように変化していくかを規定する[5]．このため要求の状態は全てのドメインの状態を管理する．図 4 に電話交換機問題における要求の状態遷移図の描画例を示す．

要求の状態遷移図作成ツール画面を開くには，プロブレム図において要求をダブルクリックする．黒丸が開始点，角の丸い長方形は状態を表し，ひとつの状態で全てのドメインの状態の組み合わせを管理する．矢印による遷移は，ドメインの状態遷移図とは異なり，入力イベントによってのみ遷移し，更に条件記述を付加する事が出来る．条件は入力イベントの記述に[ ]を追加して表記する．入力イベントはどのドメインから発生したイベントかを明確にする為，”？”を用いて”ドメイン名？イベント名”のように表記する．

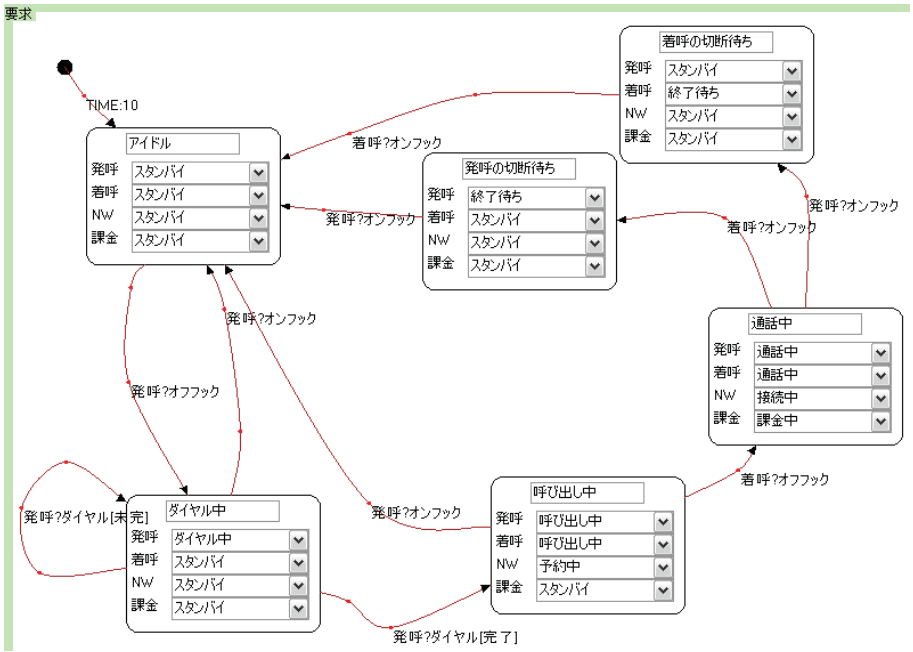

図 4　電話交換機の要求の状態遷移図

図 5 要求でのドメイン状態の選択

図 6　要求状態遷移図でのイベントの選択

要求の状態遷移図は全てのドメインの状態遷移図と関連性をもっている．要求の状態を描画すると，自動的に全てのドメインに対して状態名のリストを保持するコンボボックスが生成される．要求の状態は，状態名の入力と各ドメインの状態の選択によって規定出来る(図 5)．また，遷移のイベントに対しては，すべてのドメインの入力イベントのリストを持つコンボボックスが表示される(図 6)ので，そこから選択することによって遷移のイベントを規定できる．このように，要求の状態遷移図作成ツールはプロブレム図とドメインの状態遷移図と関連性を持つことから，状態名や入力イベントを選択することで簡単に作成できるといった入力補助機能を備えている．

3.4 要求の正しさのチェック機能

要求の状態遷移図はドメインの状態遷移図に沿って作成される．このとき，要求の状態遷移図が正しくドメインの状態遷移図の規定に合っていることがチェックされ，そうでない場合はエラーが表示される．これは，以下のエラーチェックを行う．

(1) ドメインの規定により起こりえないことを要求が想定するエラー

(1-1) 入力イベントが起こりえないエラー

要求の状態 S からイベント $D?E$(ドメイン D から E を入力)による遷移がある場合，ドメイン D の状態遷移図においては，S でのドメイン D の状態 S_d から入力 E による遷移が規定されていなければならない．

(1-2) 出力イベントが起こりえないエラー

要求の状態 S_x からイベント $D_k?E$ で状態 S_y への遷移がある場合，S_x でのドメイン D_i の状態を S_{xi}, S_y でのドメイン D_i の状態を S_{yi} とすると，D_k のドメインでは，S_{xk} から E による入力遷移とそれに続く 0 回以上の出力遷移によって S_{yk} に遷移し，それ以外のドメイン D_j では，S_{xj} から 0 回以上の出力遷移によって S_{yj} に遷移出来なければならない．

(2) ドメインの規定により起こり得ることを要求が想定していないエラー

要求の状態 Sからおこりうる全ての入力イベント(Sのドメイン状態とドメインの状態遷移図から得られる)は，Sからの遷移に現れていなければならない．

3.5 マシンの状態遷移記述の自動生成

ドメインの状態遷移図と要求の状態遷移図を作成し，それらの間の正しさをチェックしてエラーがないことを確かめた後で，マシンの仕様を状態遷移記述として自動生成することが出来る．ここでは簡単のため SDL[9]のテキスト表記を採用した．

マシンの状態遷移記述を生成するには，プロブレム図においてマシンをダブルクリックする．電話交換機の例を用いて，自動生成したマシンの状態遷移記述の抜粋を図 7 に示す．これは要求の状態遷移図において，ダイヤル中状態からの３つの遷移（発呼のオンフックによるアイドルへの遷移,ダイヤルの完了による呼び出し中への遷移,ダイヤル未完によるダイヤル中への遷移）に対して生成された仕様である．

マシンの状態遷移記述のうち，state，input，decision，next state の各文は要求の状態遷移図から生成する．output 文は，要求の状態遷移図中のドメイン状態の変化と，ドメインの状態遷移図を調べることによって生成する．例えば，ダイヤル中状態からダイヤルを入力しダイヤル完了条件が成立して呼び出し中に遷移する場合，要求の状態遷移図より，ドメインの状態は,「発呼：ダイヤル中→呼び出し中，着呼：スタンバイ→呼び出し中，NW：スタンバイ→予約中」と変化する（課金は変化しない）．発呼ドメインの状態遷移図(図 3)により，発呼にダイヤル成功を出力すればダイヤル中から呼び出し中に変わることが分かる．そこで，”output 発呼!ダイヤル成功;”が生成される．同様にして，”output 着呼!呼び出し;”と”output NW!予約;”が生成される．ここで，出力イベントは”ドメイン名!イベント名”と記述する．

```
state ダイヤル中;
    input 発呼 ? オンフック;
        nextstate アイドル;
    input 発呼 ? ダイヤル;
        decision;
        (完了):
            output 発呼 ! ダイヤル成功;
            output 着呼 ! 呼び出し;
            output NW ! 予約;
            nextstate 呼び出し中;
        (未完):
            nextstate ダイヤル中;
        enddecision;
```

図 7　自動生成されたマシンの状態遷移記述の抜粋

マシンの仕様としては，SDL 記法に加えて，C 言語プログラムを生成することも可能である．これを実行することにより，仕様の実行シミュレーションができる．

4. システムの評価

4.1 適用性について

我々は以前類似のツールを開発し UML との比較実験を行った[10]．簡単な例題(洗濯機，券売機，炊飯器，簡易 ATM)の状態遷移図を 6 名の学部学生(卒研生)に UML と提案方式で作成してもらい評価した．表 2 に平均時間と結果の状態数・遷移数を示す．時間的には，平均すると UML の方が少し短かったが，提案方式では要求だけでなくドメインの状態遷移図も作成するため，作業量が少し増加したこともある．被験者からは，提案方式ではドメインの状態遷移図を作ることが大変であったが，その後要求の状態遷移図は比較的簡単に作れたとの感想であった．また，結果の状態遷移図を見ると，UML では被験者による差が大きかったが，提案方式では差が少なかった．UML は自由度が大きいが，提案システムは個人差の少ない方式をガイドするといえる．

表 2　学生実験による比較評価の結果(状態数と遷移数は 3 人の結果)

	UML			提案システム		
	時間(分)	状態数	遷移数	時間(分)	状態数	遷移数
問題 A	65	3, 3, 3	4, 4, 5	65	3, 4, 3	5, 6, 5
問題 B	77	7, 12, 20	8, 12, 21	100	6, 6, 6	10, 10, 10
問題 C	62	5, 4, 5	11, 16, 18	58	4, 5, 4	13, 13, 14
問題 D	66	5, 5, 6	13, 19, 12	86	5, 5, 6	13, 11, 12

4.2 システム機能の有効性について

少し複雑な例として，電話交換機，電子レンジ，複雑な ATM に対して提案システムの評価を試みた(分析者は著者等)．電話交換機では不完全な要求から完全な要求に至る方式を試みた．全てのドメインの状態遷移図を作成後，

アイドル=[A?offHook]⇒ダイヤル=[A?dial]⇒通話=[A?onHook]⇒アイドル

(A は発呼ドメイン)なる不完全な要求の状態遷移図を入力した．チェック結果として，ダイヤル⇒通話と通話⇒アイドルでの出力遷移エラー(3.4(1-2))が出た．修正のため，その間着呼ドメイン(B)からの入力イベント(それぞれ B?offHook と B?onHook)を待つ状態として，呼出中と B 終了待ちを挿入した．これで，

アイドル→ダイヤル→呼出中→通話→B 終了待ち→アイドル（イベント省略）

の主ループが出来た．このチェックで，ダイヤルと呼出中で A?onHook イベントが，ま

た通話で B?onHook イベントが発生しうるとのエラーが出た.この修正として新たな遷移を追加し，さらに，dial 入力時の判定(未完，完了，ダイアルエラー等)を加えて最終的な電話交換機の要求状態遷移図が得られた．このように，システムのチェック機能を活用して，断片的な要求から徐々に完全な要求を作成することが出来た.

電子レンジの例ではもともと UML の例があり，それをもとに提案システムで作成した．UML の例では，制御部の状態は環境の状態と対応していなく，例えば，ドアーの開閉は状態の明示的な変化を起こさない設計であった．調理中にドアーを開くとどうなるかは，状態遷移図中の内部処理の変数の代入や値の判定文を調べてわかるようになっていた．一方提案システムでは，要求の状態はドメインの状態を１つに決める必要があり，ドアー開とドアー閉は別状態となるため，調理中にドアーを開ければ停止する等の本質的な振舞いが状態の遷移として明示的になった.

少し複雑な ATM の例では，要求の状態遷移図を完成させた後のチェックで，「パスワードエラー時の再入力」と「銀行での認証中」という２つの状態からキャンセルが押されたときの遷移が抜けているというエラー(3.4(2)) が出た．そこで，キャンセルによりカードを排出し「カード取出し待ち」状態へ移る遷移を追加した．この修正によって，今度は銀行のドメインで，認証中からアイドルへの出力遷移がないというエラー(3.4(1-2))が出た．つまり，銀行ドメインは，認証中からは，銀行の判断による遷移(パスワードエラー，ID エラー，認証 OK 等による遷移)のみ存在し，ATM 装置からの指示による出力遷移がなかった．そこで，銀行ドメインの状態遷移図に認証中から認証中止出力によるアイドル状態への遷移を新たに追加し，エラーをなくすことができた．これらのエラーは分析者の事前レビューでは発見されなかったもので，システムのチェック機能の有効性が確認できた.

5. おわりに

プロブレムフレームにおけるプロブレム図を作成し，それを元にドメインや要求の状態遷移図を作成し，それからマシンの状態遷移仕様を自動生成する，プロブレムフレームを用いた状態遷移分析の統合的なシステムについて述べた.

対象問題としては，組込みシステムでよく出てくる(必要な)振舞いフレームに限定した．プロブレム図作成ツールではドメインや要求の状態遷移図を作成する事を踏まえ，ドメインのイベントや状態の管理を重視し，プロブレム図の重要な概念である副問題への分解や合成といった機能については省略した．ドメインの状態遷移図作成ツールでは，プロブレム図との互換性を重視し，プロブレム図にて記述した情報から容易にドメインの状態遷移図を作成出来るようにした．要求の状態遷移図作成ツールでは，全てのドメインの状態遷移図から容易に要求の状態遷移図を作成出来るように，要求の各状態でのドメイン状態を選択できるようにした．マシンの状態遷移記述を自動生成する機能は，プロブレム図やドメインの状態遷移図，要求の状態遷移図によって分析を行った後，その結果のまとめとして，次の段階へと進んだソフトウェアの設計に有効なマシンの仕様を作成する事を目的とした．さらに，これらの図の間の整合性をチェックする機能により，ドメインや要求の記述の誤りや抜け等を自動的に検出できるようにした．これによって，複雑なシステムの分析においても，各ドメインの仕様と要求の記述を分離し，それぞれは比較的単純で理解しやすい形で記述でき，それらの合成結果としてマシンの状態遷移記述を誤りなく生成出来るようにした.

今後の課題として，まずは評価の多様化を挙げる．今回の評価実験は開発者自らシステムを使用しての考察であり，主観的であったのも事実である．今後は第三者による客観的な評価を行いたい．さらに実システムに対しての適用・評価も行っていきたい．次に，システムの機能強化として，より複雑な場合の対処が課題となる．１つは，プロブレムフレームの重要な概念である副問題への分割・合成機能の開発がある．分割や合成では，プロブレム図の分割，合成に伴って，ドメインや要求の状態遷移図の適切な分割や合成の実現が課題となる．これができれば，複雑な問題は，いくつかの比較的簡単な副問題を合成することによって自動的に得ることが出来る.別の方法は，画面のサイズ，ズーム，編集機能，自動レイアウト機能等の充実によって，複雑な図形も比較的簡単に扱えるようにすることである．今後，これらの課題への対応を進めていきたい.

参考文献

[1] OMG : UML 2.0 Superstructure Specification, http://www.omg.org/, 2004.

[2] IBM Rational ソフトウェア：http://www-06.ibm.com/software/jp/rational/

[3] JUDE/Biz：http://jude.change-vision.com/jude-web/index.html

[4] Jackson, M.: Problem Frames: Analyzing & Structuring Software Development Problems, Addison-Wesley Publishing Company, 2001

[5] 紫合治:ジャクソン法(JSP)による状態遷移設計,情報処理学会論文誌 vol.50 No12, pp.3041-3051, 2009.

[6] Alfaro, L. and Henzinger, T.A.:Interface Automata, ACM SIGSOFT FSE01, pp.109-120, 2001.

[7] Clarke,Jr., E., Grumberg O. and Peled, D.A. , Model Checking, The MIT Press, 1999.

[8] 紫合治：状態指向のステートチャート，ソフトウェア工学の基礎ワークショップ（FOSE05), pp.25-30, 2005.

[9] 若原恭，長谷川晴朗：仕様記述言語 SDL，株式会社カットシステム(1996)

[10] 大川敦,加藤大輝,紫合治:インターフェースの情報を用いた状態遷移図の作成，情報処理学会研究報告 2006-SE-151，pp.33-40，2006.