

プロブレムフレームによる 状態遷移図分析支援システム

横山 薫[†] 紫合 治^{††}

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着目し、プロブレムフレーム図を作成、それをもとにドメインや要求の状態遷移図を作成することにより要求分析を行い、さらに要求分析の次の段階であるソフトウェア設計のためのマシンの状態遷移記述を自動生成する、総合的な分析支援システムについて述べる。システムは、プロブレムフレーム図で定義したイベントをそのままドメインの状態遷移図で活用し、ドメインの状態遷移図で定義した状態をそのまま要求の状態遷移図で使うことで、これらの作成を容易にする。また、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も持つ。

A Support System for State Transition Diagrams in Problem Frames

Kaoru Yokoyama[†] and Osamu Shigo^{††}

This paper describes a total support 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、ユースケースだけでなく、シーケンス図、アクティビティ図、概念クラス図などが活用されている。これら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、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ソフトウェアによく出てくる振舞いフレームに限定し、その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含んだ分析支援システムについて述べる。

2. プロブレムフレームとは

2.1 プロブレムフレームの特徴

ソフトウェア開発の問題で中心となる目標は、現実の世界で何か有益な目的を達成するソフトウェアを作成することである。解決策とはコンピュータとシステムに存在するが、問題は現実世界に存在する。プロブレムフレームとは問題に集中し、現実世界を分析・構造化・パターン化を行ってからソフトウェア開発に取り掛かるものである。プロブレムフレームによる電話交換機の例を用いて、プロブレムフレーム図について具体的な説明を次項で行う。

2.2 プロブレムフレームと状態遷移図

図1は本論文で扱うプロブレムフレーム図を示す。2本の垂直な線が入った長方形はマシンを表し、プロブレムフレーム図において1つだけ存在する。垂直な線のない

[†] 東京電機大学院 情報環境学研究科
Graduate School of Information Environment, Tokyo Denki University

^{††} 東京電機大学 情報環境学部
School of Information Environment, Tokyo Denki University

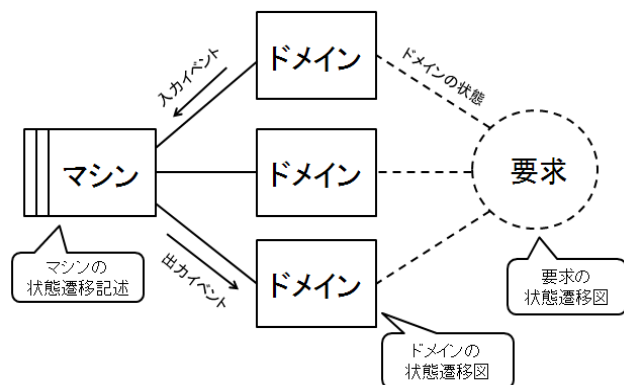


図1 プロブレムフレーム図と状態遷移

長方形はマシンを取り巻く現実世界の要素であるドメインを表す。破線の楕円はドメインに対する要求を表す。マシンとドメイン間の実線には入出力イベント関係の記述を行う。ドメインからマシンに至るイベントは、ドメインが自動的に行うもので、これを(マシンの)入力イベントと呼ぶ。マシンからドメインに至るイベントは、ドメインにとって他動的なイベントであり、これを(マシンの)出力イベントと呼ぶ。要求とドメイン間の破線には要求から見たドメインの状態を記述する。この様に、ここではマシンからはドメインの入出力イベントを参照し、また要求からはドメインの状態を参照するものとする[5]。

各ドメインには入出力イベントと状態の関係を示した状態遷移図(ドメインの状態遷移図)をドメインプロパティとして規定する。要求にはドメインの状態の組みの変化の様子を示した状態遷移図(要求の状態遷移図)を規定する。マシンの仕様として、ドメインからの入力イベントに対し、ドメインへの必要な出力イベントを規定した状態遷移(マシンの状態遷移記述)を規定する。

2.3 プロブレムフレームによる電話交換機問題図の例

図2はプロブレムフレーム図による電話交換機の例を示す。電話交換機の問題とは、電話交換機というマシンが発呼、着呼、ネットワーク、課金という要素を通じ、ダイヤルや呼び出し、課金のON/OFFの切り替え、ネットワークの接続を正しく行うものである。発呼とは電話を掛ける側の電話機を表すドメインで、スタンバイ、ダイヤル中、呼び出し中、通話中、終了待ちの状態があり、発呼ドメインと要求間の破線に沿って記述する。また、発呼ドメインが自動的に行う入力イベントには、オフフック、オンフック、ダイヤルがあり、発呼ドメインがマシンから受け取る入力イベントには

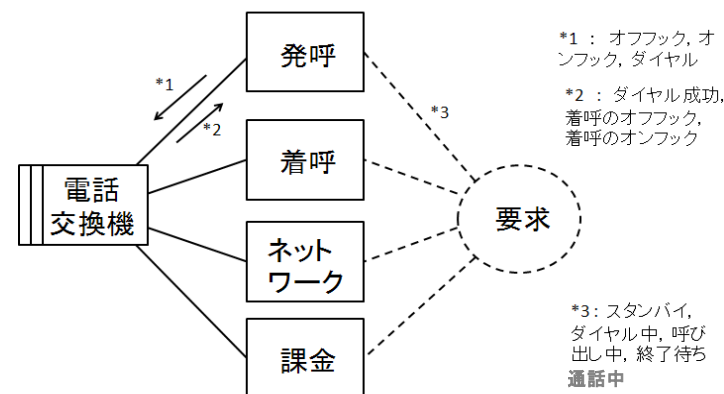


図2 電話交換機システムのプロブレムフレーム図の例

ダイヤル成功、着呼のオフフック、着呼のオンフックがあり、電話交換機と発呼ドメイン間の実線に沿って記述する。

2.4 副問題

プロブレムフレームの特徴は現実世界の分析・構造化である。問題が大きく複雑な構造である場合、より小さく単純な副問題に分解する事で理解しやすくなる。副問題はすでに検討を始めている大きな問題の一部であり、図3は図2において多々ある副問題のうち、ネットワークの接続に対する課金の要求に特化した副問題である。ネットワークの接続に対する課金の要求とは、ネットワークが接続されている間だけ課金をONにし、それ以外の状態ではOFFにしておくという振る舞いである。関係のあるドメインのみ抜粋し副問題へ分解するが、発呼ドメインや着呼ドメインはこの副問題とは無関係であるから表示されない。

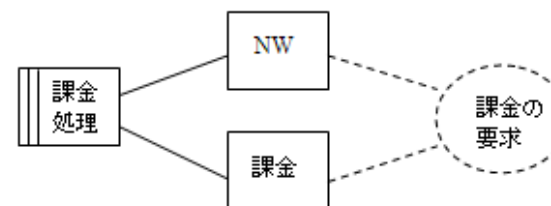


図3 課金処理に限った副問題図の例

3. 支援システム

本システムはプロブレムフレーム図の作成や、副問題への分解や副問題同士の合成を行う事が出来る。更に、プロブレムフレーム図を元に各ドメインの状態遷移図を作成し、ドメインの状態を使った要求の状態遷移図を作成する機能をもつ。また、これらの状態遷移図からマシンの大まかな仕様となるマシンの状態遷移記述を自動生成する事が出来る。システムの開発環境には Microsoft 社の Visual C#2005 Express Edition を用いた。

3.1 プロブレムフレーム図作成ツール

図 4 はプロブレムフレーム図作成ツールの基本画面である。各種の要素とインターフェースを用いて、左側のキャンバス上にプロブレムフレーム図を作成していく。

図形の描画の他、イベントの管理、副問題への分解、副問題同士の合成、ファイルのセーブロード機能を備えている。

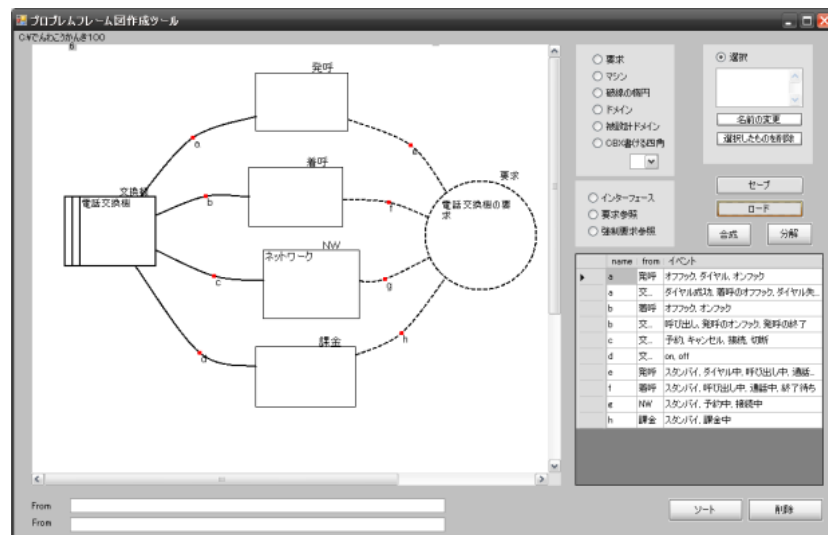


図 4 システムの基本画面

3.1.1 問題図の作成

ツールではプロブレムフレームのルールに沿った作成を行えるような機能を備えている。例えばプロブレムフレーム図においてマシンや要求は各問題図に 1 つしか存在する事が出来ないため、2 個目を作成しようとした場合、図 5 のようなエラーメッセージが発生する。

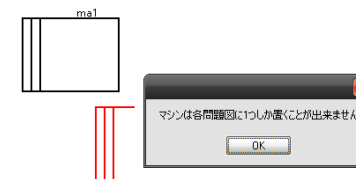


図 5 マシン要素重複登録時のエラーメッセージ表示

3.1.2 イベント管理表

図 6 は電話交換機問題における、基本画面右下のイベント管理表を示す。まずインターフェースの線を描画すると、イベント管理表においてそのインターフェースに関連する入力イベントと出力イベントを記述する為の空の行が 2 行自動生成される。マシンとドメイン間のインターフェースの場合、イベント管理表の from がドメイン名の行には入力イベントを記述し、from がマシン名の行には出力イベントを記述する。また、要求とドメイン間のインターフェースの場合、イベント管理表の 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 6 ではイベントが空白の行は削除されている)。

	name	from	イベント
▶	a	発呼	オフフック, ダイヤル, オンフック
	a	交...	ダイヤル成功, 着呼のオフフック, ダイヤル失...
	b	着呼	オフフック, オンフック
	b	交...	呼び出し, 発呼のオンフック, 発呼の終了
	c	交...	予約, キャンセル, 接続, 切断
	d	交...	on, off
	e	発呼	スタンバイ, ダイヤル中, 呼び出し中, 通話...
	f	着呼	スタンバイ, 呼び出し中, 通話中, 終了待ち
	g	NW	スタンバイ, 予約中, 接続中
	h	課金	スタンバイ, 課金中

図 6 イベント管理表

3.1.3 副問題への分解

分解したい電話交換機の問題図を表示し分解ボタンを押すと、図 7 のような分解フォームが出現する。残したいドメインを選択し、OK ボタンを押すと図 8 のように副問題の図が得られる。選択したドメインが残り、イベント管理表においては関係のあ

る行だけが表示される。逆に、いくつかの副問題を指定し、合成ボタンを押すことによって合成問題を得ることが出来る。なお、現状では副問題の分解・合成は、後に述べる状態遷移図には反映されない。

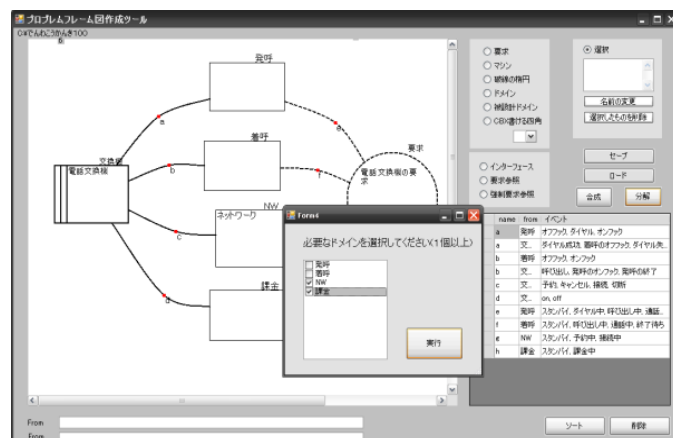


図 7 副問題への分解指示

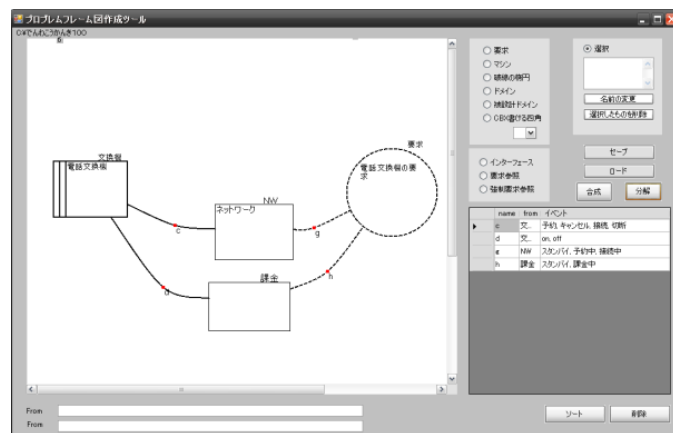


図 8 副問題の生成

3.2 ドメインの状態遷移図作成ツール

プロブレムフレームにおけるドメインは、それぞれいくつかの状態をもつ。状態の情報はイベント管理表でドメインと要求間に引かれる破線に対する項目として記述されている。ドメインの状態遷移図とは、各ドメインがどのようなイベントによって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、プロブレムフレーム図と互換性を持ちながら作成する事が出来る。

図 9 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。

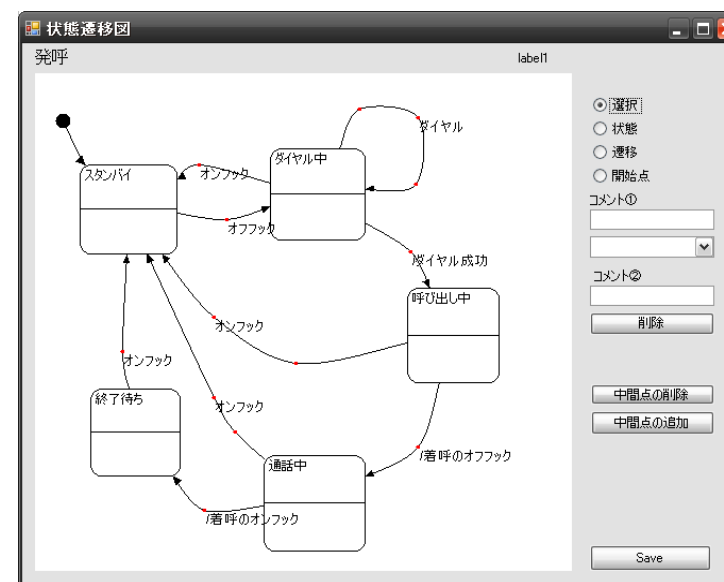


図 9 発呼ドメインの状態遷移図

ドメインの状態遷移図作成ツール画面を開くには、プロブレムフレーム図において作成したいドメインをダブルクリックすればよい。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フレーム図のイベント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

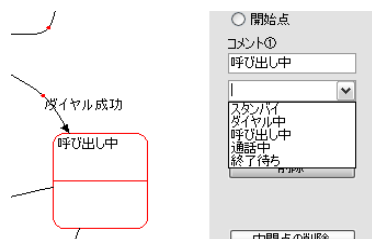


図 10 状態名の選択

選択することによって状態名を指定できる (図 10). 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フレーム図のイベント管理表に反映される.

発呼ドメインの振る舞いは次の通りである. スタンバイ状態時にオフフックを行う事によりダイヤル中へと遷移し、ダイヤル中状態においてダイヤルを行うと同じ状態に遷移し、ダイヤル中状態、通話中状態、終了待ち状態においてオンフックを行うとスタンバイ状態へと遷移する. これらは発呼ドメイン自身が自働的に行う入力イベントによる遷移である. 発呼ドメインの他動的なイベントである出力イベントによる遷移としては、ダイヤル中状態からダイヤル成功イベントによる呼び出し中への遷移、呼び出し中から着呼のオフフックイベントによる通話中への遷移、また通話中から着呼のオンフックイベントによる終了待ちへの遷移がある.

3.3 要求の状態遷移図作成ツール

プロブレムフレーム図を作成し、全てのドメインに対してドメインの状態遷移図を完成させる事により、要求の状態遷移図を作成する事が出来る. 要求の状態遷移図とは、あるドメインの入力イベントに対して全てのドメインの状態の組がどのように変化していくかを規定する[5]. このため要求のひとつの状態は全てのドメインの状態を管理する. 図 11 に電話交換機問題における要求の状態遷移図の例を示す.

要求の状態遷移図作成ツール画面を開くには、プロブレムフレーム図において要求をダブルクリックすればよい. 黒丸が開始点、角の丸い長方形は状態を表し、ひとつの状態ですべ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違い、入力イベントによってのみ遷移し、更に条件を付加する事が出来る. 条件は入力イベントの記述に[]を用いて表記する. 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.

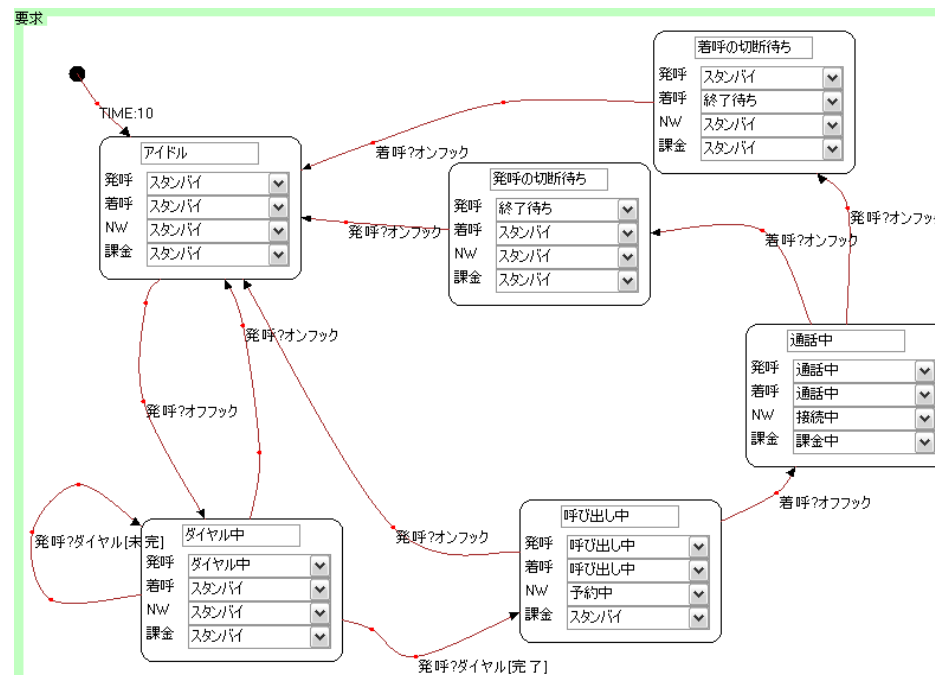


図 11 電話交換機の要求の状態遷移図

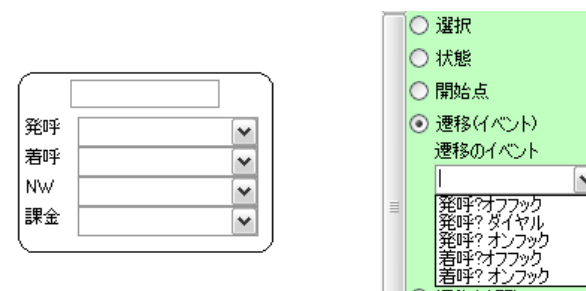


図 12 要求の状態遷移図の入力補助機能

要求の状態遷移図はプロブレムフレーム図と全てのドメインの状態遷移図と関連性をもっている。状態を作成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自動生成される(図 12 左)。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。また、遷移イベントである全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も自動生成される(図 12 右)。このように、要求の状態遷移図作成ツールはプロブレムフレー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だけで簡単に記述が出来るといった入力補助機能を兼ね備えている。

要求の状態遷移図における、アイドル状態から呼び出し状態までの振る舞いは次の通りである。アイドル状態における各ドメインの状態は(発呼:スタンバイ, 着呼:スタンバイ, NW(ネットワーク):スタンバイ, 課金:スタンバイ)である。アイドル状態において発呼のオフフックという入力イベントによってダイヤル中状態へと遷移する。ダイヤル中状態における各ドメインの状態は(発呼:ダイヤル中, 着呼:スタンバイ, NW:スタンバイ, 課金:スタンバイ)であるが、発呼ドメインの状態がスタンバイからダイヤル中へと遷移している事が判る。この時発呼ドメインの状態遷移図において、スタンバイ状態からオフフックという入力信号によってダイヤル中状態へと遷移する、といった記述が無くてはならない。つまり入力イベントが発生したドメインは、開始状態からその入力信号によって次状態へと遷移する記述が必要である。

次にダイヤル中状態から呼び出し中状態への遷移をみていく。ダイヤル中状態から発生する遷移は3本ある。1つは発呼のオンフックによるスタンバイ状態への遷移である。残りの2つの遷移は発呼のダイヤルという同一の入力イベントによる遷移ではあるが、条件によって遷移する次状態が変わる。ダイヤル中状態において、発呼のダイヤルという入力イベントに”未完”という条件が付加された場合、開始状態であるダイヤル中状態へと遷移する。一方、発呼のダイヤルという入力イベントに”完了”という条件が付加された場合、呼び出し中状態へと遷移する。呼び出し中状態における各ドメインの状態は(発呼:呼び出し中, 着呼:呼び出し中, NW:予約中, 課金:スタンバイ)となる。

3.4 マシンの状態遷移記述の自動生成

プロブレムフレーム図やドメインの状態遷移図、要求の状態遷移図は要求の明確化を目的とした図であるが、本システムではこれらの図を全て作成する事によってマシンの振る舞いを自動生成する事が出来る。このマシンの振る舞いの仕様をマシンの状態遷移記述と呼ぶ。図 13 のように、マシンの状態遷移記述とは、ドメインからの入力に反応し、ドメインへの必要な出力を規定したものである。要求の状態遷移図とドメインの状態遷移図を参照し自動生成することが出来る。

マシンの状態遷移記述は SDL 記法[6]を参考に作成する。SDL 記法とは通信システムの動作を正確に仕様化する事を目的としたものであり、テキスト記述と図式記述の

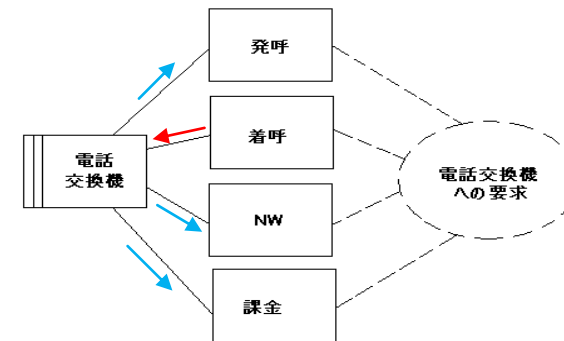


図 13 マシンの状態遷移記述の振舞い

2通りがあるが、本システムではテキスト記述を採用している。

図 14 はマシンの状態遷移記述の全体構成である。システム記述、ドメイン記述、シグナル記述はそれぞれ存在するシステム全体のファイル名、存在する全てのドメイン名、すべてのイベント名を記述する。主な記述はプロセス記述であり、状態遷移記述によってマシンの仕様を表す。

```

マシンの状態遷移記述:=
  システム記述
  ドメイン記述
  シグナル記述
  プロセス記述

システム記述:= system ファイル名
ドメイン記述:= domain 略ID: ドメイン名 [略ID: ドメイン名]...
シグナル記述:= signal 略ID 入力/出力 信号
プロセス記述:= process マシン名[状態記述]...

状態記述:=
  state 状態名;
  [入力記述]...
入力記述:=
  input ドメイン名?イベント名;
  {判定記述|出力記述}
判定記述:=
  decision;
  {(条件):出力記述}...
  enddecision
出力記述:=
  [output ドメイン名|イベント名]...
  nextstate 状態名;
  
```

図 14 マシンの状態遷移記述の全体構成

マシンの状態遷移記述を生成するには、プロブレムフレーム図においてマシンをダブルクリックすればよい。

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におけるプロセス記述の抜粋を図 15 に示す。抜粋は要求の状態遷移図において、ダイヤル中状態からダイヤル完了によって呼び出し中へと至り、呼び出し中状態から着呼のオフフックによって通話中状態へと至る遷移である。

```
process 電話交換機
state ダイヤル中;
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;
state 呼び出し中;
input 着呼? オフフック;
output 発呼! 着呼のオフフック;
output NW! 接続;
output 課金! on;
nextstate 通話中
```

図 15 自動生成されたマシンの状態遷移記述の抜粋

マシンの状態遷移記述のうち、state, input, decision, nextstate 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、呼び出し中状態から着呼のオフフックを入力して通話中に至る遷移では、要求の状態遷移図より、ドメインの状態は「発呼: 呼び出し中→通話中、着呼: 呼び出し中→通話中、NW: 予約中→接続中、課金: スタンバイ→課金中」と変化する。着呼ドメインの状態遷移図により、着呼のオフフックによって呼び出し中から通話中に変換することが分かる。発呼ドメインの状態遷移図(図 9)より、発呼を呼び出し中から通話中に遷移させるには、着呼のオフフックイベントを出力すればよいことが分かるので、"output 発呼! 着呼のオフフック;"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。同様にネットワークと課金のドメインの状態遷移図を調べることで、"output NW! 接続; output 課金! on;"が生成される。

マシンの状態遷移記述の自動生成においては、ドメインの状態遷移図と要求の状態遷移図の整合性について以下のようなチェックを行う。

- ・ 要求の遷移に書かれた入力イベントが、要求の状態で規定されたドメインの状態からは起こり得ない。
- ・ 要求の状態で規定されたドメインの状態から起こり得る入力イベントが、要求の遷移に書かれていない。
- ・ 要求の遷移で規定されたドメインの状態変化(入力イベント処理後の状態変化)が、ドメインの出力イベントによる遷移ではできない。

図 16 に整合性チェックのエラーメッセージの例を示す。

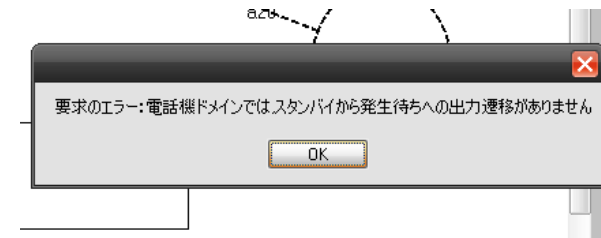


図 16 要求の整合性チェックによるエラーメッセージ出力

4. 評価

本システムを使用して、3つの例題(簡単な電話交換機、自動販売機、ATM)について分析を行った。各々の比較は表 1、表 2、表 3 の通りである。

ATM のドメイン数は 8 と、電話交換機と自動販売機の 2 倍であった。ATM は複雑な問題の例として作成したが、ドメイン数が多いほどマシンを通じてやりとりされる入出力信号は増え、要求の状態遷移図における状態数や遷移数も多くなった。表 3 に示す通り、ATM で生成されたマシンの状態遷移記述の行数は 169 行と多いことから、ATM に要求されるマシンの仕事内容が他と比べてかなり複雑な事が判る。

従来の方によってマシンの仕様をいきなり間違いなく作るのは大変だが、要求とドメインの状態遷移図を分割して作成し、それからマシンの状態遷移記述を自動生成する事によって複雑なものも作成する事が出来た。

すなわち複雑な内容でも、本システムを利用する事によって、比較的簡単なドメインの状態遷移図と要求の状態遷移図に分けることにより、理解しやすく且つ誤り無く作成する事が可能となることが確かめられた。

表 1 ドメインの状態遷移図要素数

ドメインの状態遷移図	状態数	入力信号数	出力信号数	ドメイン数
電話交換機	14	9	12	4
自動販売機	11	8	11	4
ATM	37	15	48	8

表 2 要求の状態遷移図要素数

要求の状態遷移図	状態数	遷移数
電話交換機	6	10
自動販売機	5	12
ATM	14	22

表 3 自動生成されたマシンの状態遷移記述行数

マシンの状態遷移記述	行数
電話交換機	64
自動販売機	74
ATM	169

5. おわりに

本論文ではプロブレムフレームを作成し、それを元にドメインや要求の状態遷移図を作成し、全てを参照してマシンの状態遷移図を自動生成する、プロブレムフレームを用いた総合的なシステムについて述べた。

プロブレムフレーム図作成ツールでは要求やドメインの状態遷移図を作成する事を踏まえ、イベントの管理に重きを置いた仕様とし、プロブレムフレームの重要な概念である副問題に纏わる分解や合成といった機能を実装した。ドメインの状態遷移図作成ツールでは、プロブレムフレーム図との互換性を重視し、プロブレムフレー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仕様とした。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仕様とした。マシンの状態遷移記述を自動生成する機能は、プロブレムフレー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要求分析を行った後、発展として要求分析段階より次の段階へと進んだソフトウェアの設計に有効な情報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、ドメインの記述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ている。これによって、複雑なシステムの分析においても、各ドメインの仕様や要求は比較的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無く生成出来るようにしている。

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、3.1(3)で述べた副問題の分割・合成機能の充実がある。分割や合成に伴って、状態遷移図の適切な分割や合成が課題である。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く所存である。

参考文献

- [1] 渡辺政彦, 飯田周作, 石田哲史, 山本修二, 浅利康二: UML 動的モデルによる組み込み開発 分析・設計・実装・テスト, 株式会社オーム社
- [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] Michael Jackson, プロブレムフレーム ソフトウェア開発問題の分析と構造化, 株式会社翔泳社(2006)
- [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(2009)
- [6] 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)