

通信システムの視覚的仕様記述支援法

更科克幸† 安藤津芳† 太田正孝† 高橋薫‡

†高度通信システム研究所 ‡東北大学

通信システムの要求仕様を、視覚的仕様記述手法を使い、LOTOSの形式的記述に生成する方法を提案する。本論文で提案した視覚的仕様記述手法は、仕様作成者が、従来の設計仕様図式記述方法を複数使い、自然な解釈に基づく視覚的表現形式を、多面的に組み合わせる事によってLOTOS仕様を記述する手法であり、従来の設計仕様記述手法を使用しているので、ドキュメント作成等の工程削減、設計者との要求確認も、容易に行うことができる。

A Visual method for Communication Systems Specification

Katsuyuki Sarashina† Tsuyushi Ando† Masataka Ohta† Kaoru Takahashi‡

†AIC System Laboratories

6-6-3, MinamiYoshinari, Aoba-ku, Sendai-shi, Miyagi, 989-32 JAPAN

‡Tohoku University

2-2-1, Katahira, Aoba-ku, Sendai-shi, Miyagi, 980 JAPAN

ABSTRACT In this paper we propose a method for making LOTOS Specifications of the communication system, using visual development technique. These visual methods enable to describe LOTOS Specifications, using natural visual forms which are composed versatilly. They also enable to reduce costs of making documentations and support designers for making confirmations easily.

1. はじめに

コンピュータシステム、通信システムにおいて形式仕様記述技法(Formal Description Techniques = FDTs)は、プログラミングを行う以前の要求仕様作成時点でFDTs定義に従い記述し、仕様の曖昧性を排除するために開発された技法である。

形式仕様記述技法の1つであるLOTOS^[1]は、ISOで開発されたFDTsの1つであり、要求仕様解析に有効視されている数学的枠組をもっており、計算機による検証方法が研究されている^[6]。つまりLOTOSフォーマットで要求仕様を記述すれば、従来のウォーターフォール型ソフトウェア開発パラダイムの中で最も工数を費やしているテスト、検査の工程を軽減し、手戻りなどの開発効率の低下を初期の段階で押さえる事ができる。

しかし、現在のLOTOSフォーマットでの記述は、高級プログラム言語のようにテキスト型のままであり、仕様記述者が使用したり理解したりするものには程遠く、直接的な利用については計算機の支援が無い状況である。

本論文で提案する、視覚的仕様記述手法は、従来から、そのシステム構築のために使われていた複数の設計仕様記述方法を要求仕様の視覚的な入力環境として用いて、それらの情報を有機的に結合・統合することで、形式仕様記述言語への変換を行う方法である。

次章では、主に通信システムで、使用されている設計仕様定義の視覚的な記述方法と、現在LOTOSで提案されているグラフィカル表現方法^[4]との差異、問題点を整理し、要求仕様記述の際の、視覚的な表現を使用した多面的な仕様記述手法について述べる。

2. 要求仕様図式表現

2.1 従来の設計仕様図式表現

ある要求をモデル化するために次の3種類の考え方が、特に大規模な設計仕様を記述する上で一般的である^[7]。

●空間モデル

[時間の概念を持たず、複雑な複合概念に基づく構造を記述する.]

例として図1に空間モデル記述の代表である機能ブロック図を示す。機能ブロック図は、構成要素の骨格(ex.: Process A, B, C)に該当し主にプロセスの機能分担などを表すことに使用される。

●時間モデル

[順序関係、状態遷移などの時間的概念を記述する.]

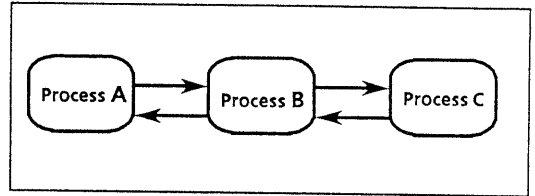


図1. 機能ブロック図

例として図2に順次フロー図、図3に状態遷移図を示す。これらの図式表現は上記空間モデルで記述された構成要素内の1要素(ex.: Process A)の詳細化に使用され、状態ごとの動作、時間の流れを表すことに使用される。

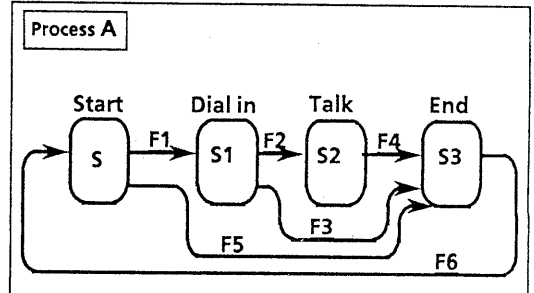


図2. 順次フロー図

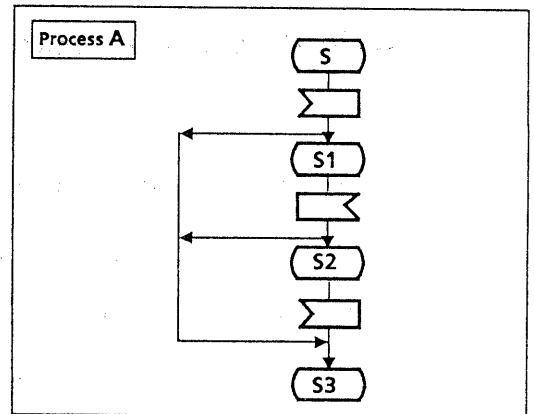


図3. 状態遷移図

●時空間モデル

[空間概念と時間概念をとともに意識し時間モデルによって記述された対象が空間に複数存在し相互に通信し合うものを記述する.]

例として図4にプロセス間フロー図を示す。この表現は、並列処理、同期処理などの相互に影響し合う際(ex.: Process AとProcess B間など)に必要なデータを送受、時間の流れを表すことに使用される。

このようなモデル化の方法は、要求をインプリメントするために、要求を分割して整理し、異なっ

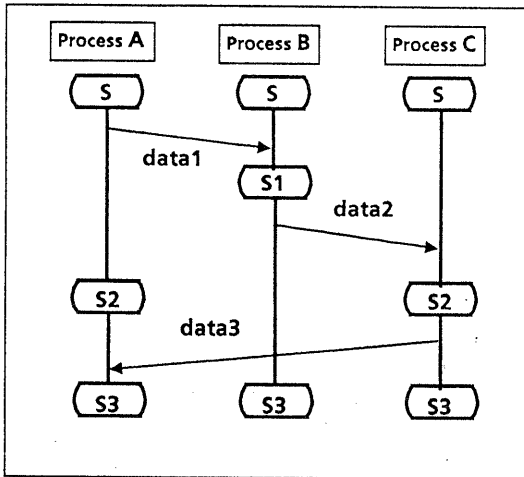


図4. プロセス間フロー図

た側面から更に分割・合成しながら仕様を作成する方法である。

2.2 グラフィカルLOTOS仕様表現

形式仕様記述言語LOTOSにおいて、提案されているグラフィカル表現²⁾を使い、2.1項で示した仕様を図5に示す。

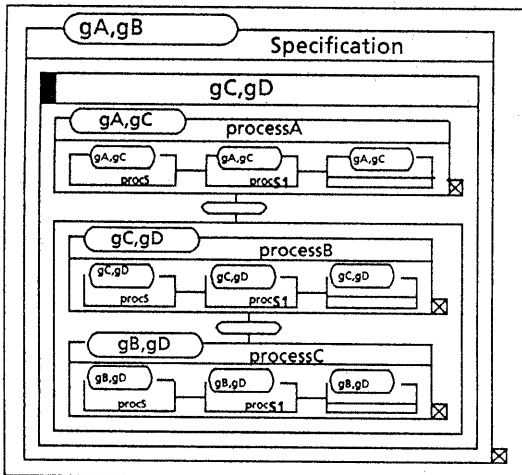


図5. グラフィカルLOTOS表現

従来の要求仕様記述と同様な目的で作成されているLOTOSのグラフィカル表現は、従来の水平投影法(木構造的表現方法)とは異なり、次のような利点がある。

1)従来の水平投影法では、木構造の方向が1方向に限られていたが、グラフィカルLOTOS表現は垂直投影法を採用しているために上下左右・垂直に記述することが可能で広域的な処理の流れ、局所的な流れの方向を意図的に変えることが出来、仕様の全体像が直感的に描きだせる。

2)垂直投影法の採用により、LOTOSの特徴である並列、選択、順次などの表現が描ける。

3)グラフィカルLOTOSの表現は、テキスト型LOTOSと基本的に1対1に対応するため、グラフィカルLOTOSからテキスト型LOTOS、テキスト型LOTOSからグラフィカルLOTOSへの相互変換が比較的容易にでき、変換に伴ってのセマンティクスが損なわれない。

グラフィカルなLOTOS表現には、上記の様に利点があるが、以下に示すような問題も存在している。

1)垂直投影法において、グラフィカルシンボルの大きさが可変で内包的なシンボルの変化によって内包するグラフィカルシンボル自体の大きさが変化する。このため、レイアウトに対する計算機的支援が難しい。

2)水平投影法では、階層レベルが同じものを同一の高さで表現でき、階層レベルの位置、深さ、順次性が直感的に把握できるが、垂直投影の場合は分かりにくい。

3)グラフィカルな表現から計算機処理がしやすいテキスト型LOTOSへの変換が簡単に出来る分、従来、仕様記述者の中で使われていた構造図、状態遷移図、コントロールシーケンス図などの概念分析が、一つの画面上に全て存在するために、仕様記述者に対して要求の分析、合成という作業が更に複雑な形で実施することを強いられる。

4)グラフィカルなシンボルが、単体のシンボルではなく複合体となって表現されるため、シンボル自体の組み合わせによって、意味解釈が異なる。

このように、従来の図的モデル定義要素全てを同一フォーマット上に記述することによって、仕様記述者が直感的に見るデータ量が膨大になる。また、実際のシステムの本質的な構造、振舞いが形式仕様記述言語の記述定義に束縛されたモデル表現となるために、直感的な理解性が低下する。更に、仕様記述者がLOTOSに対して、十分な知識がない限り記述できないという問題がある。

3. 図的表現を使った多面的な仕様記述手法

最終的に要求仕様を仕様記述者が記述でき、要求仕様記述の段階で仕様検証、自動プログラミングが出来る事が、最も望ましい。

しかし、前節で指摘したように、現状の形式仕様言語のままでは、記述することも、解読することも仕様記述者にとっては、容易ではない。また、交換システムなどの大規模なシステムを構築する際には、多くの人との共同作業が必要であり、仕様記

述者と設計者、あるいは設計者間での共通な表現形式が必要になるが、現時点では、形式仕様記述言語が共通の表現形式になることは考えにくい。

そこで、従来の設計仕様表現をそのまま用いて、入力した情報から、形式仕様言語LOTOSでの、仕様作成の可能性を調べる。まず、従来の設計仕様記述表現がどのような側面から、どのような情報を獲得しているかを分析する。次に、形式仕様記述言語LOTOSが、どのような情報によって記述され、従来のどの設計仕様記述表現から変換に必要な情報を収集できるかを示す。

3.1 仕様記述情報

従来から使用されている設計仕様記述表現方法を、第2節で説明したモデル化に基づき、空間モデルでは、機能ブロック図、時間モデルでは、状態遷移図、時空間モデルでは、プロセス間フロー図を用いて説明する。

従来の設計仕様表現方法と仕様記述情報を表1に示す。

表1. 従来の設計仕様表現方法と仕様記述情報

仕様記述情報	FBD	STD	PSF
プロセス名	○	○	○
入出力の関係		○	○
プロセス間信号シーケンス			○
接続関係	○		
単純処理シーケンス		○	○
分岐処理シーケンス		○	
状態処理シーケンス		○	
データ名		○	○

FBD:機能ブロック図 STD:状態遷移図
PSF:プロセス間フロー図

FBDでは、プロセス名とプロセス間接続関係のみが表現でき、STDでは、単体プロセスの名前、入出力関係、内部の処理シーケンス、データ名が表現される。PSFは、プロセス名、データ名の他に、プロセス毎に記述された入出力の関係から、プロセス間信号シーケンスが表現できる。

次にLOTOS仕様作成において必要な情報が、どの仕様記述表現形式で得られるかを表2に示す。

LOTOS記述に必要な、process-identifier(プロセス名)、gate-identifier(インタラクションポイント)は、FBDで収集ができ、インタラクションポイントの方向性、behaviour-expression(動作)は、STDで収集ができる。そして、プロセス毎の接続

表2. LOTOS仕様記述情報と獲得情報

LOTOS記述情報	FBD	STD	PSF
process-identifier	○	○	○
gate-identifier	○	○	○
! , ?		○	○
behaviour-expression		○	
sort-identifier		○	○
value-identifier			○

FBD:機能ブロック図 STD:状態遷移図
PSF:プロセス間フロー図

関係やsort-identifier(データ名)は、PSFから収集できる。

この2つの表からは、従来の仕様記述表現から得られる情報のみで、LOTOS仕様表現が必要としている情報が作成できることを示している。

つまり、空間モデル・時間モデル・時空間モデルそれぞれの仕様記述の情報があれば、その情報を有機的に結合することにより、1つの整理された要求仕様が生成される。

3.2 図的仕様記述表現の相互変換

図的仕様記述表現の相互変換は、仕様の変更、追加、削除に関しての矛盾の指摘、見落とし防止に必要である。

表1、表2でも分かるように、ある仕様記述情報について、一つの表現形式が既に与えられていれば、新しく他の表現を用いて、別の側面を記述する際には、既に与えられている情報を利用して、仕様記述を容易にすると同時に両表現の矛盾が回避される。

更に、ある記述表現形式(ex.:プロセス間フロー図)から、他の表現形式の存在の可能性(ex.:プロセス間のシーケンス追加など)が指摘されれば、その可能性を提示し、更に要求仕様の詳細化を促すことが可能になる。

3.3 多面的図式仕様からLOTOS仕様の生成

具体例を通して、多面的図式仕様からLOTOS仕様の生成を考察する。

特に、今回は交換システムの場合で従来から使われている要求仕様記述表現(機能ブロック図、状態遷移図、プロセス間シーケンス図)を使い説明する。

なお、データ定義(ADT)に関しては、抽象度が高く、現在もデータの図的仕様記述が見当たらないこと、今回対象とする交換システムの信号シーケンスに限定すれば、標準的にデータ名が決定しているの、あらかじめLOTOS仕様にライブラリ

の形で付与しておき、必要に応じて作成すると仮定する。

1) 機能ブロック図からLOTOS仕様への写像

機能ブロック図は、要求対象となるもの、機能負担を示すものであり、要求仕様の骨格が表現される。この要求仕様の骨格は、LOTOSの動作(振舞い)の骨格として、写像される。

簡単な例の機能ブロック図を図6に、LOTOS仕様への写像イメージを図7に示す。

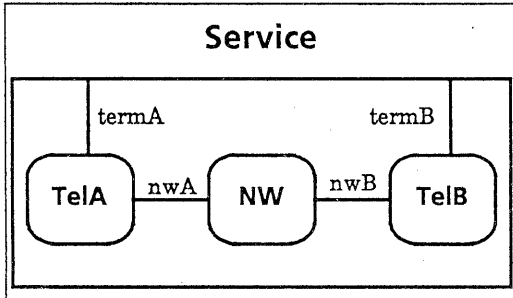


図6. 機能ブロック図

```

specification Service[ termA,termB]:noexit
behaviour hide nwA,nwB in
  TelA[termA,nwA]
  [nwA]
  NW[nwA,nwB]
  [nwB]
  TelB[termB,nwB]
  
```

図7. LOTOS仕様への写像イメージ

TelA, TelBは、発信、着信電話機、NWは、交換機、termA, Bは、人が電話機に操作を行うアクセスポイント、nwA, Bは、それぞれのプロセスの接続アクセスポイントを表現している。

2) 状態遷移図からLOTOS仕様への写像

機能ブロック図で明示された個々の機能ブロック(プロセス)の順次的な動作を、内部の状態を変化することによって記述してゆく。ここでの記述表現では、LOTOSのプロセスの詳細化に使用され、状態から次状態までのシーケンスが詳細プロセスに写像される。また、状態のループなどの表現は、LOTOSの再帰的記述に該当する。

図8は、発信電話機(TelA)の状態遷移図を示しており、端末未使用状態(idle)から、アクセスポイント(termA)を介してオフフック(offhook)信号を受信し、状態が遷移する事などを記述している。この状態遷移図から、LOTOS仕様への写像イメージを図9に示す。

この部分の状態遷移図は、個々の機能ブロックを個別に記述するため、全ての機能ブロックを記述することが原則となる。今回使用している記述スタイルは交換仕様記述で一般的に使用されてい

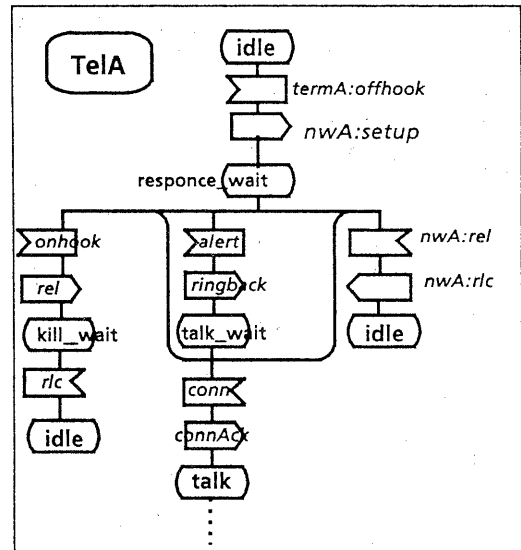


図8. 状態遷移図

```

Process TelA[termA,nwA]:noexit :=
  idle[termA,nwA]
where
  process idle[termA,nwA]:noexit :=
    termA?offhook;nwA!setup;
    response_wait[termA,nwA]
  endProc
  process
    response_wait[termA,nwA]:noexit :=
      termA?onhook;nwA!rel;
      rlc_wait[termA,nwA]
      [] nwA?alert;termA!ringback;
      talk_wait[termA,nwA]
      [] nwA?rel;termA!bt;
      kill_wait[termA,nwA]
    endProc
  process rlc_wait[termA,nwA]:noexit :=
    nwA?rlc;TelA[termA,nwA]
  endProc
  process talk_wait[termA,nwA]:noexit :=
    nwA?conn;nwA!connack;
    talk[termA,nwA]
    [] termA?onhook;nwA!rel;
    rlc_wait[termA,nwA]
    [] nwA?rel;termA!bt;
    kill_wait[termA,nwA]
  endProc
  ...
endProc
  
```

図9. LOTOS仕様への写像イメージ

るSDL/GR[4]の表現形式を用いている。SDLからLOTOSへの変換方法も研究されている[5]。

3) プロセス間フロー図からのLOTOS変換

プロセス間フロー図は、機能ブロック図で分割され、状態遷移図で個々に記述された内部動作によって発生する出力イベントをその他の機能ブロックとシーケンスで合成し、シーケンスの整合性、インタフェースの確認を行う。

例のTelA, TelB, NW間のフロー図を図10に示す。

LOTOS仕様記述でも、個々の状態遷移図からの変換イメージの合成、データの属性設定が行われ、プロセスの順次、包含関係が写像される。

LOTOS仕様への写像イメージを図11に示す。

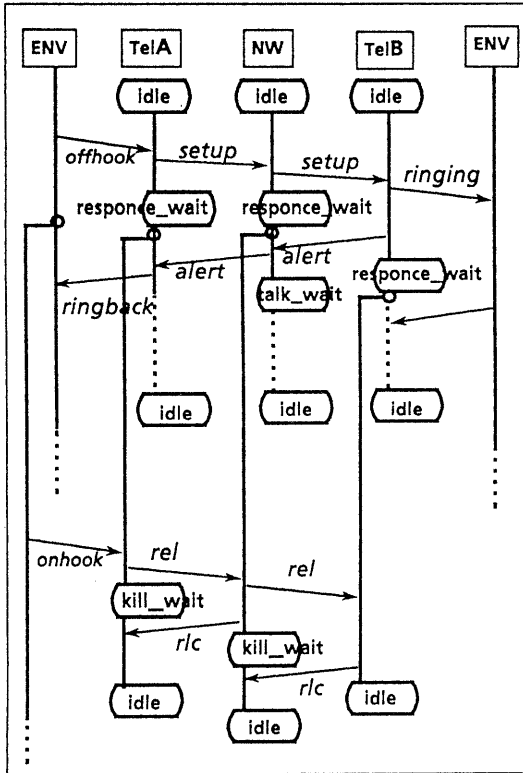


図10. プロセス間フロー図

以上述べてきたように、必ずしも厳密ではない図的表現形式を含む個々の設計仕様記述を組み合わせることによって、要求を順序立てて分析し、それぞれの得意な側面を記述することにより、非形式的な要求仕様記述から形式的仕様記述を生成しうることを提示した。

このような、手法を用いることによって、

- 1) 仕様設計者自身がその専門領域に從來から使用している図的仕様記述を継続して使える。
- 2) 仕様設計者とLOTOS専門家とが協力して、要求を仕様定義に変換する場合でも、主として半形式的な仕様表現を介して、共通の情報を得ることができる。

などの特徴が上げられ、汎用的な支援システム、形式的仕様記述が必ずしも使い易くない状況を見れば、仕様記述者に受け入れられやすいと考えられる。

4. 支援環境の構成

視覚的な要求仕様記述支援の構成は、図12に示すように、従来のエディタ群で入力された情報を

```

specification Service[ termA,termB]:noexit
type external_signal is
  sorts ex_sig
  opns offhook,onhook,ringing,ringback:->
  ex_sig
endtype
type internal_signal is
  sorts signal
  opns
  setup,alert,conn,connAck,rel,rlc:->signal
endtype
behaviour
  hide nwA,nwB in
  TelA[termA nwA]
  [[nwA]]
  Nw[nwA,nwB]
  [[nwB]]
  TelB[termB,nwB]
where
  process TelA[termA,nwA]:noexit =
  idle[termA,nwA]
  where process idle[termA,nwA]:noexit =
  termA?x:ex_sig[x=offhook];
  nwA!setup;
  response_wait[termA,nwA]
  endProc
  process
  response_wait[termA,nwA]:noexit =
  termA?x:ex_sig[x=onhook];
  nwA!rel;
  kill_wait[termA,nwA]
  [] nwA?x:signal[x=alert];
  termA!ringback;
  talk_wait[termA,nwA]
  endProc
  process kill_wait[termA,nwA]:noexit =
  nwA?rlc;TelA[termA,nwA]
  endProc
  :
endProc
process NW[nwA,nwB]:noexit =
  idle[nwA,nwB]
  where process idle[nwA,nwB]:noexit =
  nwA?x:signal[x=setup];nwB!setup;
  response_wait[nwA,nwB]
  process response_wait[nwA,nwB]:noexit =
  nwA?x:signal[x=rel];nwB!rel;
  nwA!rlc;NW[nwA,nwB]
  []nwB?x:signal[x=alert];nwA!alert;
  talk_wait[nwA,nwB]
  :
  endProc
endProc
process TelB[termB,nwB]:noexit =
  idle[termB,nwB]
  where
  process idle[termA,nwA]:noexit =
  nwB?x:signal[x=setup];
  termB!ringing;
  response_wait[termB,nwB]
  endProc
  process response_wait[termB,nwB]:noexit =
  termB?x:ex_sig[x=offhook];
  :
  [] nwB?x:signal[x=rel];
  termA!ringbacknwB!rlc;
  TelB[termB,nwB]
  :
  endProc
endSpec

```

図11. LOTOS仕様への写像イメージ

フィルターで選別し、その情報を基に形式仕様言語の作成を行う形式である。

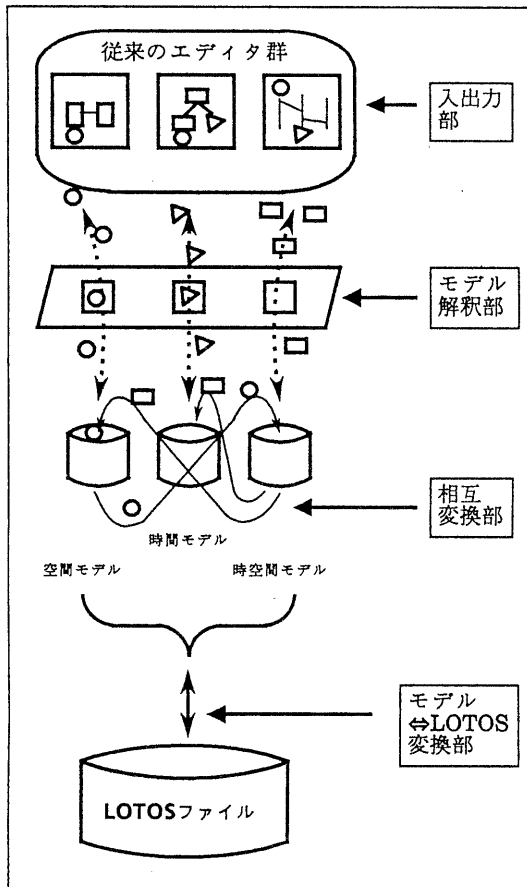


図12. 視覚的な要求仕様記述支援環境

以下に構成要素の機能を説明する。

- 1) 入出力部
従来の視覚的な設計仕様記述環境を、そのまま視覚的な要求仕様記述環境として用いる。
- 2) モデル解釈部
空間モデル、時間モデル、時空間モデルの3種類モデルの側面から、仕様記述情報を収集する。
- 3) 相互変換部
視覚的な設計仕様記述システム間の、共通情報の相互変換を行い、仕様の変更、追加、削除などに関して、影響が出ると判断できる場合、影響する支援システムに変更情報等を通知する。
- 4) LOTOS変換部
上記の1), 2)で得られた要求仕様を有機的に結合させ、LOTOS仕様記述に変換する。また、LOTOS仕様の情報をそれぞれの、設計仕様記述の要素に分解し、逆方向の変換も可能とする。

5. まとめ

本論文で提案した、視覚的仕様記述手法は、仕様検証機能があるLOTOS仕様作成を、支援する。

仕様作成の支援環境の入力支援としては、自然な解釈に基づく視覚的表現形式を、提供することが望ましいが、現在の要求仕様の図式表現は、対象となるシステムに依存しない、抽象度の高い表現の為に、仕様設計者が、容易に理解することが出来ない。

本論文の入力支援環境は、従来、システム固有に設計仕様作成の時、使用する図的表現方法を使用している。この図的表現方法は、設計者が設計仕様を解析、確認する為であり、論理立てて、色々な側面から記述することが可能である。

LOTOS仕様記述は、要求モデルの複合体の為、一度に全ての情報を記述するのは、容易ではない。しかし、視覚的表現形式を、多面的に組み合わせ、獲得した情報を、有機的に結合することによって、形式仕様記述言語(LOTOS)を生成する方法は、仕様作成者に混乱が少なく、容易に仕様記述が可能で、直感的な理解性も向上すると考える。

謝辞

本研究を進めるにあたり、御指導、御支援を頂いた東北大学の野口正一教授、ならびに白鳥則郎教授に深謝致します。また、研究をする機会を与えてくださった同研究所の緒方秀夫常務、有益な御意見を頂いた同研究所の諸氏に感謝致します。

【参考文献】

- [1] ISO, "Information processing systems - Open Systems Interconnection - LOTOS - A formal description technique based on the temporal ordering of observational behaviour", ISO 8807, 1989
- [2] ISO/IEC, "Information processing systems - Open Systems Interconnection - LOTOS - A formal description technique based on the temporal ordering of observational behaviour Proposed Draft Addendum 1: G-LOTOS", ISO/IEC 8807/PDAD1, 1989
- [3] 更科・他, "GLOER-G-LOTOSエディタの試作", 情報処理学会第42回全国大会, 1991
- [4] CCITT, "Functional Specification and Description Language (SDL)", Recommendation Z.100, (CCITT), 1988
- [5] Ando, Ohta, Takahashi: "Translation from SDL to LOTOS". SDL '91: Evolving Methods, pp95~104 (North-Holland), 1991
- [6] Yamano, Jokanovic, Ando, Ohta, Takahashi, "Formal Specification and Verification of ISDN Services", 信学技報SAT91-25, 1991
- [7] 榎本, "ソフトウェア工学ハンドブック", オーム社, 1986