

SDL グラフィックエディタの設計と製作†

宗 森 純†† 水 野 忠 則††

本論文では SDL グラフィックエディタの設計方針と実現方法について述べる。SDL グラフィックエディタは CCITT が情報通信システム用に勧告している形式的仕様記述言語 SDL の図式表現である SDL/GR を編集するためにワークステーション上に実装したものである。本エディタは情報通信システムの日本語仕様書等のドキュメントを正確かつ迅速に作成および修正し、従来の手書き、もしくは専用のエディタを使用しない場合と比較して生産効率を上げることが目的としている。そのため、情報通信システムで頻繁に起こる出力シンボルと入力シンボル間の複数のプロセスにまたがる信号の流れを明確にし、仕様記述時における誤りの早期発見に適した、関連プロセスの同時表示機能を備えている。実装に際しては、情報通信システムの仕様の容易な記述と CCITT 勧告準拠および日本語支援を設計方針とした。この方針のもとに、図式表現 SDL/GR とテキスト表現 SDL/PR との相互変換、文字列からの入力法、視覚的な編集、格子状の画面、複数のプロセスグラフの合成および同時表示、記述の自動的な検査等の機能を実現した結果、実用に十分に耐え、また仕様書の品質を高めることが分かった。

1. ま え が き

情報通信システムを設計するためには、システムの仕様を明確に規定し、実装においてあいまいさが残らないようにする必要がある。

近年、情報通信システムの発展に伴い、大規模なシステム事例が増え、多くの人が協力して開発を行わなければならないとなっている。そのため、システムの仕様の規定があいまいであっては、統一的な開発ができず、システムの開発が困難となる。

また、情報通信システムでは複数の計算機や端末が、お互いに並行して処理を行いつつ、必要に応じて同期をとって目的とすべき処理を遂行する同期処理が行われているので、そのシステムの仕様は複雑となり、従来使用されてきたフローチャート等で正確に記述することが困難となっている。

このため、情報通信システム用の並列処理が規定できる形式的記述技法の研究が各方面で進められている^{1)~3)}。一方、国際標準の場においても、ISO (International Organization for Standardization) において Estelle⁴⁾ と LOTOS⁵⁾、また、CCITT (International Telegraph and Telephone Consultative Committee) において SDL (Specification and Description Language)⁶⁾ の標準化が進んでいる⁷⁾。

これらの形式的仕様記述言語の中で、SDL は視覚的に理解しやすい図式表現をもち、既に交換機分野

において仕様記述に広く使用されている。しかし、これらは多くの場合、手書き、もしくは日本語ワードプロセッサを用いて記述されている^{8),9)}。また、ミニコンピュータや汎用計算機とグラフィックディスプレイを組み合わせたグラフィックエディタについての開発も行われてはきたが、これらはまだ日本語の支援や操作性等において実用に供するためには検討すべき課題も多いのが現状である¹⁰⁾。

一方、ワークステーションの進展に伴って、日本語機能およびビットマップディスプレイやポインティングデバイスを利用した、より高度なマンマシンインタフェースを有するグラフィックエディタの実現が可能となってきた。

我々は実使用を念頭におき、情報通信システムの仕様を日本語で記述可能とするとともに、単一のプロセスの記述に限定されていた従来のエディタとは異なり、誤りの早期発見に適した、複数の同期プロセスの同時表示機能を備え、ドキュメントを効率的に作成することを目的とした SDL グラフィックエディタをワークステーション上で開発した。

以下、本論文では、第2章において SDL の特徴について触れ、第3章では SDL での記述をワークステーション上に実現した SDL グラフィックエディタの設計方針と実現形態について述べる。第4章では機能の特徴について説明する。第5章では本グラフィックエディタで実現した機能について評価を行う。

2. SDL

SDL は元来、電話交換機システムの仕様を記述するために CCITT で開発された言語であり、1976 年

† Designing and Implementing an SDL-based Graphic Editor on a Workstation by JUN MUNEMORI and TADANORI MIZUNO (Information Systems and Electronics Development Laboratory, Mitsubishi Electric Corp.).

†† 三菱電機(株)情報電子研究所

に勧告されて以来、改訂作業が4年に一回行われ、1984年にはデータ通信プロトコル記述に対応するべく、様々な機能が追加され Z. 100 から Z. 104 が勧告された。

SDL の言語仕様は、システムのブロック構造を示すブロックダイアグラムと、システムの動作内容を示すプロセスグラフから構成されている。また、表現方法として、意味的に等価な図式的な表現 (SDL/GR) とテキスト的な表現 (SDL/PR) がある。

(1) ブロックダイアグラム

通常、電話交換機システムのプログラムでは、全体のシステムを呼処理ブロック、システム管理ブロック、障害処理ブロック等に機能単位で分割して記述する。また、これらのブロックは互いに連携を取りながら動作している。SDL では、これに対応して記述の対象とするシステムを、機能的にブロック単位に分け、ブロック相互、またはシステムの外部環境とはチャンネルを通じて結ぶ。

実際のシステムプログラムにおいて、それぞれの機能ブロックはいくつかのプロセスで記述されているように、SDL においても各ブロックの動作はいくつかのプロセスによって記述される。各プロセスはコネクションの設定や開放に応じて動的に生成し消滅する。

ブロックやプロセスはさらに細かく分割することができるので、複雑なシステムも記述可能となっている。

(2) プロセスグラフ

プロセスグラフは、プロセスの動きを拡張状態遷移機械のモデルの観点から、図1に示すようなシンボルを用いてグラフ的に記述するものである。

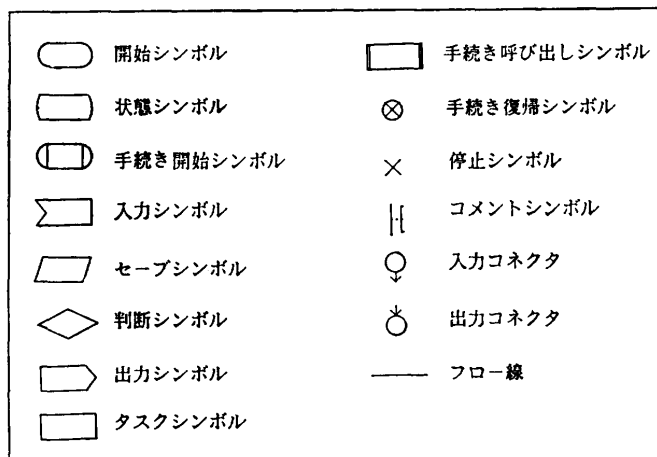


図1 SDL グラフィックエディタがサポートする SDL/GR シンボル
Fig. 1 Supported symbols of SDL graphic editor.

すなわち、プロセスの動作手順に関して、まず開始シンボルによってプロセスの始まりを規定する。プロセスは、通常入力シンボルの指定によって入力されるまで待ち、入力されると次の状態に移移する。遷移の際にはタスクシンボルを用いて実行内容を規定する。そして、出力シンボルを用いて出力を規定し、判断による選択的な動作が起こる。判断には判断シンボルを用いる。そして出力には出力シンボルを用いる。プロセスは停止シンボルによって、遷移を止め、消滅することを指定する。

このように SDL は、通常のフローチャートが単一のプロセスでの入力データから出力データへの加工方法について記述しているのに対し、情報通信システムの基本的な動作である、複数プロセスがお互いにデータをやりとりしながら、状態遷移の考えのもとに、容易にシステムの動的な仕様を記述できる点に特徴を有している。

3. SDL グラフィックエディタの設計方針と実現形態

3.1 設計方針

SDL グラフィックエディタは、情報通信システムの開発に必要なドキュメントを効率的に作成するための SDL/GR 用図形編集ソフトウェアである。設計にあたっては下記の点に留意した。

- (1) 情報通信システムの仕様の容易な記述
- (2) CCITT 勧告準拠
- (3) 日本語支援

(1)とする理由：情報通信システムの設計者が苦にならないマンマシンインタフェースで仕様を記述できるようにするため。

(2)とする理由：SDL は既に CCITT から勧告されている国際標準の言語であり、広く交換機分野において使用されてきている。仕様書の流通性を考えると、できる限り勧告に従うことが望ましいため。

(3)とする理由：

実際に作成する仕様書中の SDL/GR のシンボル内の文章は日本語で書く必要があるため。

3.2 実現機能

これらの設計留意点に従って、我々は次のように実現した。

- (1) ワークステーション上での実現

従来、開発されてきた SDL 用のエディタは

汎用計算機を利用していたため、応答性や操作性に問題があった。我々は、32ビットアーキテクチャを有し、20インチビットマップディスプレイ、マルチウィンドウ、2ボタンマウス等の入出力装置を持つマルチメディアエンジニアリングステーション MELCOM ME-1000 シリーズ¹¹⁾上で、SDL グラフィックエディタを開発した。

(2) 記述対象

SDL グラフィックエディタは、CCITT 勧告の中で、実際の仕様記述に使用されることの多いプロセスグラフを中心に、プロセスグラフを補佐するツリー図も支援する。プロセスグラフは、プロセス図と手続き図からなる。ツリー図は、プロセスグラフの階層関係を示す。

なお、ブロックの相互関連に関しては、別途開発のデータフロー図エディタ¹²⁾を利用する。

(3) 設計仕様書作成手順

SDL グラフィックエディタは、図2に示すように入力、編集、検査および出力の4フェーズを介して、情報通信システムの設計仕様書作成に対して支援を行う。すなわち、入力フェーズでは、日本語入力が可能な3種類の入力方法を持ち、編集フェーズでは入力したデータをもとに追加や削除等の修正を行った後、検査フェーズにおいて構文上の誤りを検出し、最後に出力フェーズで、必要な部分の図式を印刷する。

4. SDL グラフィックエディタの特徴

SDL グラフィックエディタにはSDL/GRのシンボルの追加や削除といった一般的なエディタの機能に加え、情報通信システムの日本語仕様書等のドキュメントを効率的に作成するために、以下の特徴を持たせた。

4.1 入力フェーズ

本グラフィックエディタは、SDL/GRおよびSDL/PRによる通常の入力方法に加え、新たに文字列を先に入れるSDL/GR補助入力方法を支援している。

(1) 通常の入力方法

視覚的に内容の理解が容易であるSDL/GR入力方法および初期の大量入力に適しているSDL/PR入力方法を共に支援する。

SDL/GR入力方法による場合は、操作メニューやシンボルメニューをマウスで指示し、シンボルの種類

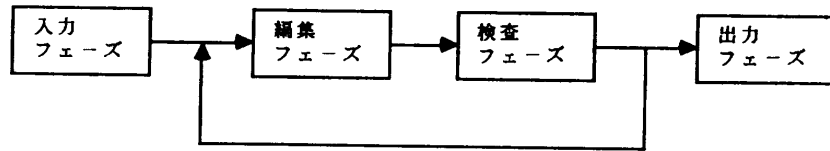


図2 SDL グラフィックエディタによる設計仕様書作成手順
Fig. 2 Description procedure of design specification using SDL graphic editor.

と挿入位置を決め、そしてキーボードによって文字を入力する。SDL/PR入力方法による場合は、ワークステーションに装備されているテキストエディタを用いてテキストファイルを作成した後、これをSDL/GRに変換する。図3に、SDL/GRおよびそれをSDL/PRに変換したCCITT勧告のトランスポートプロトコル¹³⁾の例を示す。SDL/PRをSDL/GRに変換することも可能である。

これら二つの入力方法は、どちらかと言えば、下書きがあったり、既にあるツリー図やプロセスグラフの修正に適した入力方法である。

(2) SDL/GR 補助入力方法

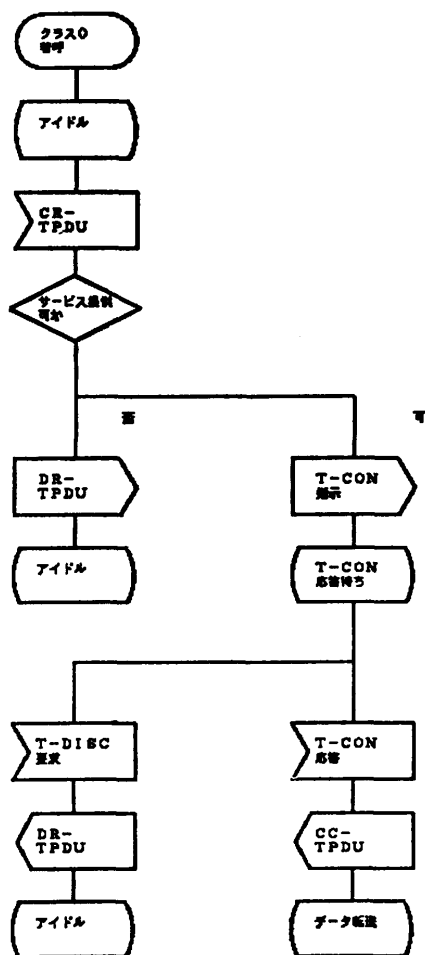
より高速にプロセスグラフを作成可能とするとともに、下書きがない、言わば発想の段階での作成支援のために、キーボードによる文字入力から直接プロセスグラフを作成するSDL/GR補助入力方法を新たに開発した。

この入力方法は通常の日本語ワードプロセッサで用いられている文書入力方法を基にプロセスグラフを作成可能とするものである。すなわち、図3の事例を本入力方法で作成する場合、まず図4(a)のようにシンボルの中に入る言葉を、作成しようとするシンボルごとにスラッシュ記号(/)で分離し、入力する。ここで文字編集が終わると図4(b)のように、タスクシンボル(矩形)を用いて、図式化される。次に図4(c)に示すように、所望のシンボルに修正する。図中@マークはシンボル内での改行を示す。ここでSDL/GR補助入力方法を用いた入力フェーズから抜け、編集フェーズに入る。この図では「サービス提供可か」のところで分岐するので、それより下にある複数のシンボルを編集機能の移動コマンドを用いて同時に移動する(図4(d))。最終的にフロー線と分岐の条件を記述して完成させる。

4.2 編集フェーズ

(1) 視覚的な図式編集

従来、図式を編集する場合、ファイルリストからその図式のファイル名をキー入力等で選択して編集作業



PROCESS クラス0@着呼;
 STATE アイドル;
 INPUT CR-@TPDU;
 DECISION サービス提供可か;
 (否):
 OUTPUT DR-@TPDU;
 NEXTSTATE アイドル;
 (可):
 OUTPUT T-CON@指示;
 NEXTSTATE T-CON@応答待ち;
 ENDDCISION;
 STATE T-CON@応答待ち;
 INPUT T-DISC要求;
 OUTPUT DR-@TPDU;
 NEXTSTATE アイドル;
 INPUT T-CON@応答;
 OUTPUT CC-@TPDU;
 NEXTSTATE データ転送;
 ENDPPROCESS クラス0@着呼;

図3 SDL/GR およびそれに対応する SDL/PR
 Fig. 3 Correspondence of SDL/GR to SDL/PR.

を行う場合が多かったが、本エディタではツリー図に視覚的に表示されているプロセスシンボルや手続き開始シンボルをクリックすることによって、直接その対象となる画面を表示し、編集可能とした。同時に3個の図式を表示および編集可能である。図5にこの例を示す。この例ではツリー図（裏側）で全体のモジュール構成を見ながら図3で示したプロセスグラフ（表側）を編集しているところである。

画面全体の大きさは全体の把握と詳細部分の拡大のため3段階に変更可能とした。

(2) 格子状の画面

PAD, HCP, SPD といった構造化言語用のエディタでは、行単位で編集が可能であった。しかしながら、SDL/GR による図式は、フロー線の結合が起こるためループ等の複雑な構造が存在し、行単位の編集が不

可能である。したがって、シンボル挿入時の自動的な段下げや、削除時の自動的な段上げを実現しようとすると、その影響を検出し処理するために、エディタの構造が複雑になりがちである¹⁴⁾。そこで編集画面を格子状に限定し、エディタの構造の簡素化を図った。格子の数は縦22個そして横8個に限定している。

格子状に仕切られた長方形の部分をパーティションと呼び、パーティションごとにシンボルを一つ表示可能とした。シンボルはCCITT 勧告に従い、縦と横の比をほぼ1:2とし、同じ形のシンボルは同一の大きさとした。パーティションの大きさは同一なので異なるシンボルの外形もほぼ同一となり、全体的な流れが見やすくなっている。シンボルに関する文字は原則的にシンボル内に記入した。シンボル内には従来の使用実績から12文字（6文字×2行）まで入力できるよ

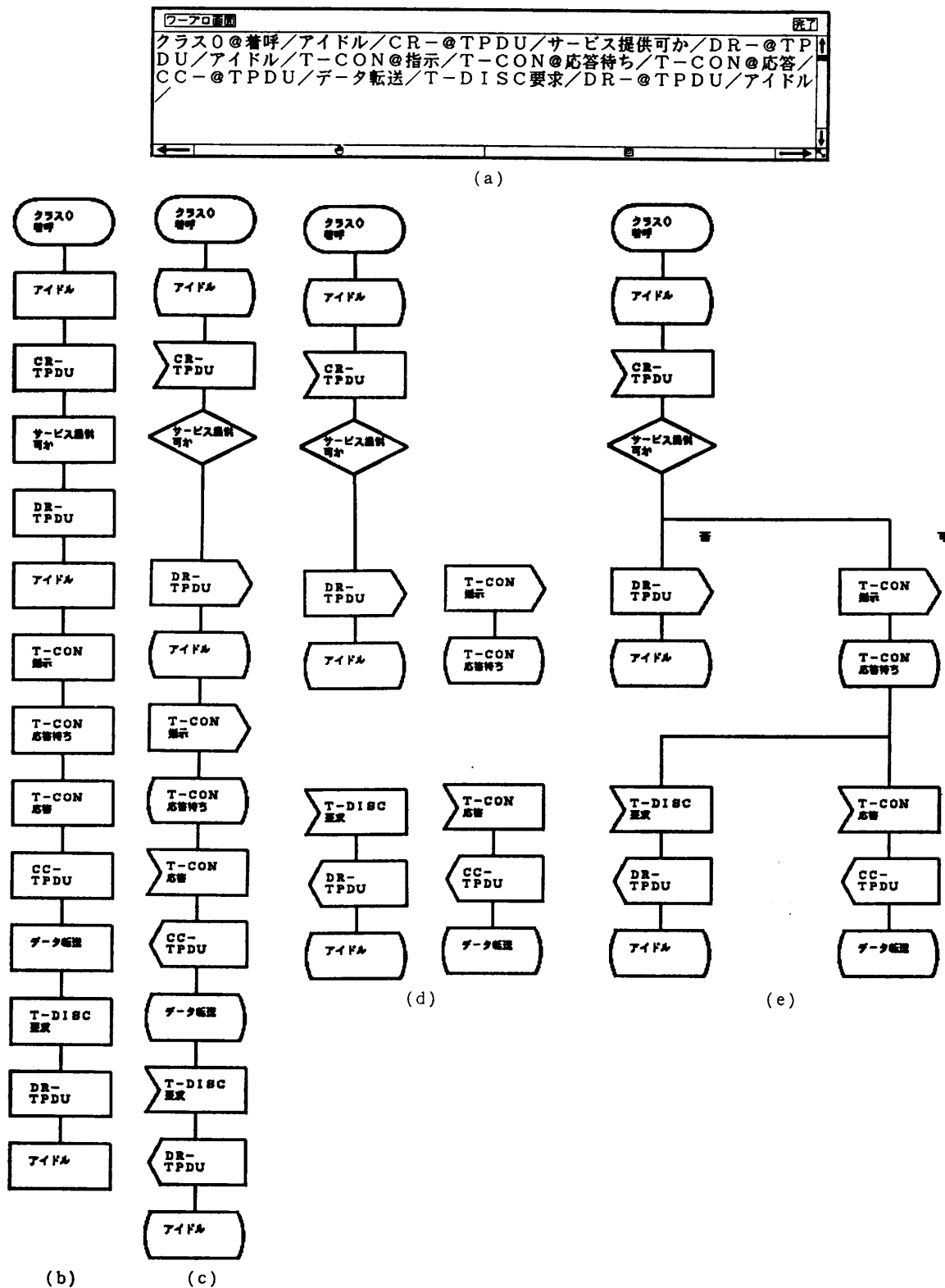


図 4 SDL/GR 補助入力方法手順例

Fig. 4 An example of the procedure with SDL/GR assistant input method.

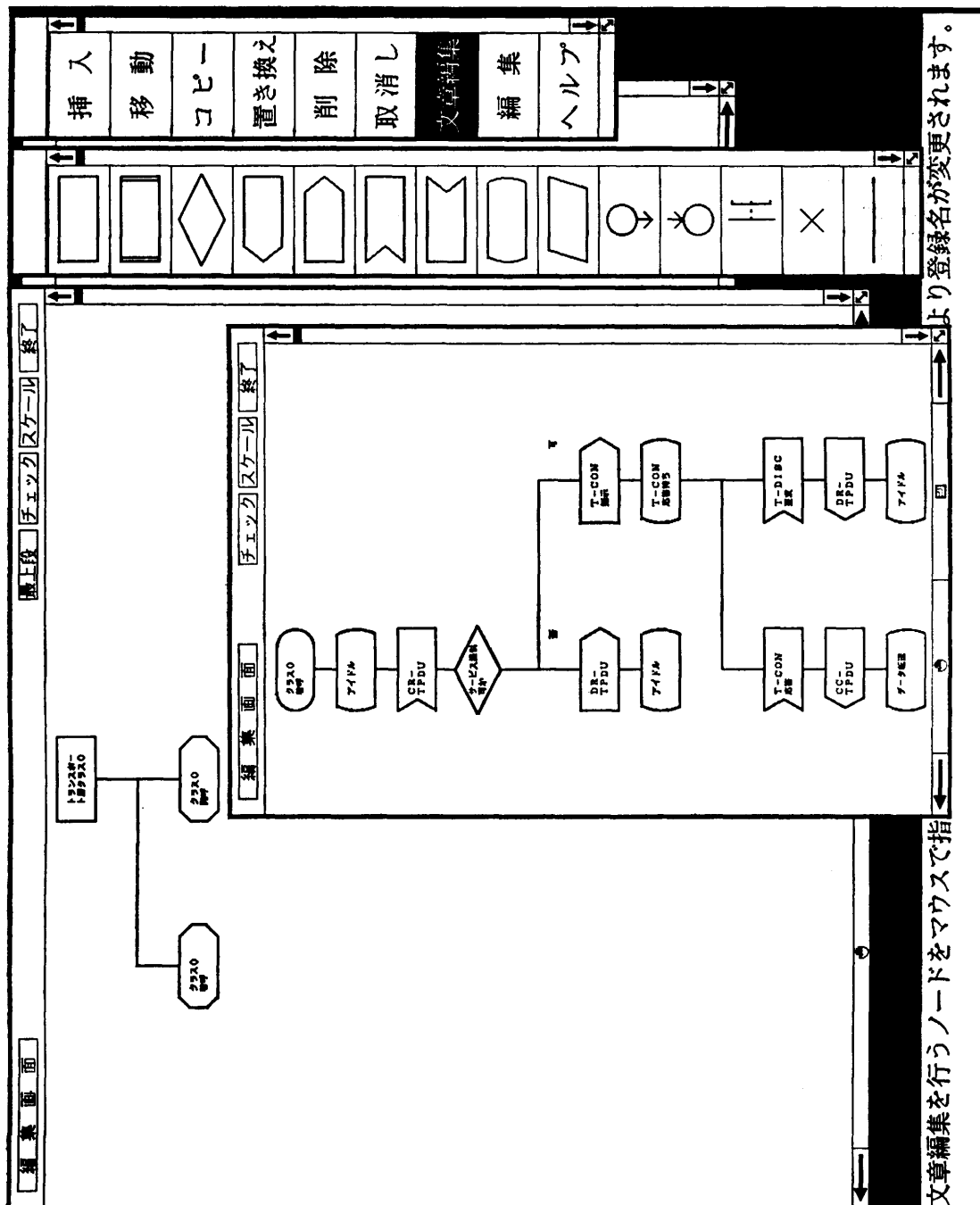


図 5 視覚的編集の例
Fig. 5 An example of visual editing.

うにした。この文字はシンボル外にはみだしてシンボルに重なることはない。シンボル内に入りきれない文字はシンボルの横にあるコメント部に最大 48 文字まで記述可能である。

編集はすべてパーティション単位で行う。このパーティションに関する情報をパーティションの位置、パーティションに表示されているシンボルの種類、

パーティションに入力された文字データおよびフロー線の入力方向および出力方向に分けてパーティションファイルに保存した。

このパーティションファイルを利用することにより、エディタの構造が簡単になったほか、シンボルの挿入による自動的な段下げ (図 6) や、複数のシンボルの一括移動や削除、コピー等が短時間で処理可能と

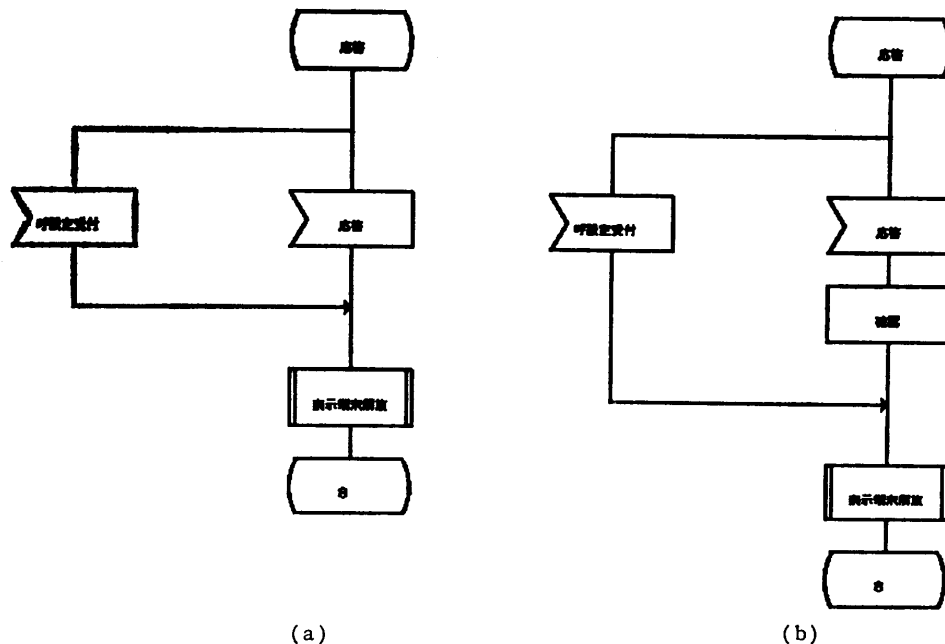


図 6 シンボルの挿入による自動的な段下げの例

Fig. 6 An example of automatic step down by symbol insertion.

なり、応答性が向上した。図 6 (b)では入力シンボルの「応答」の下にタスクシンボルで「確認」を入力したため図 6 (a)と比較すると、結合しているループの矢印以下が一段自動的に下がっている。

4.3 検査フェーズ

編集終了後、計算機上で2段階の検査を行う。すなわち、作成した図の記述上の誤りを発見するための検査と要求が正しく記述されているかの検査を行う。

(1) 作成した図の自動的な検査

フロー線がシンボルに繋がっていない等の記述上の物理的な誤りや、状態シンボルの後に入力シンボルがないといった CCITT 勧告に適合しない誤りを検出するために作成した図の検査を自動的にを行い、その誤り箇所を明示する。チェックの主な項目を表 1 に示す。この項目に適合しない場合はパーティションが反転して誤りであることを示す。

(2) 関連するプロセスグラフの同時表示による検査

仕様書の信号の流れや処理の流れが要求どおり記述されているかを検査するため、複数のプロセスグラフを同時表示し、これらの流れを追いややすくする。

異なるプロセスに信号が渡っていく同期プロセスを検査する場合、あるプロセスから別のプロセスへの信号の流れを出力シンボルもしくは入力シンボルとそれ

表 1 自動的な検査の項目例

Table 1 Check list of SDL syntax.

誤りの項目
- 文章が全シンボルに入力されていない。 - プロシジャシンボルの下にプロセスシンボルがある。 - 開始、タスク、手続き、判断、出力、入力および入力コネクタの各シンボルの下にシンボルがない。 - 判断シンボルの下に2つ以上の分岐がない。 - 同じ分岐の条件を持つ判断シンボルがすぐ上にある。 - 出力コネクタシンボルに対応する入力コネクタシンボルが複数ある。 - 状態シンボルのすぐ下のシンボルが入力シンボル、セーブシンボルでない。 - 最終の状態シンボルに対応する状態シンボルがない。 - 手続き図で復帰シンボルがない。

と同名の入力シンボルもしくは出力シンボルを呼び出して同時表示し、これらの間に信号線を描画することによって追いややすくした。同期している図の呼出しはマウスで出力もしくは入力シンボルをクリックした場合、これらに関する情報をもったファイルをアクセスし、同名の相対するシンボル（出力シンボルに対しては入力シンボル、入力シンボルに対しては出力シンボル）を持つプロセスグラフを選択することによって実現した。

同一プロセス内の手続き図を検査する場合、そのプロセス図と、それから呼び出される手続き図を同時表

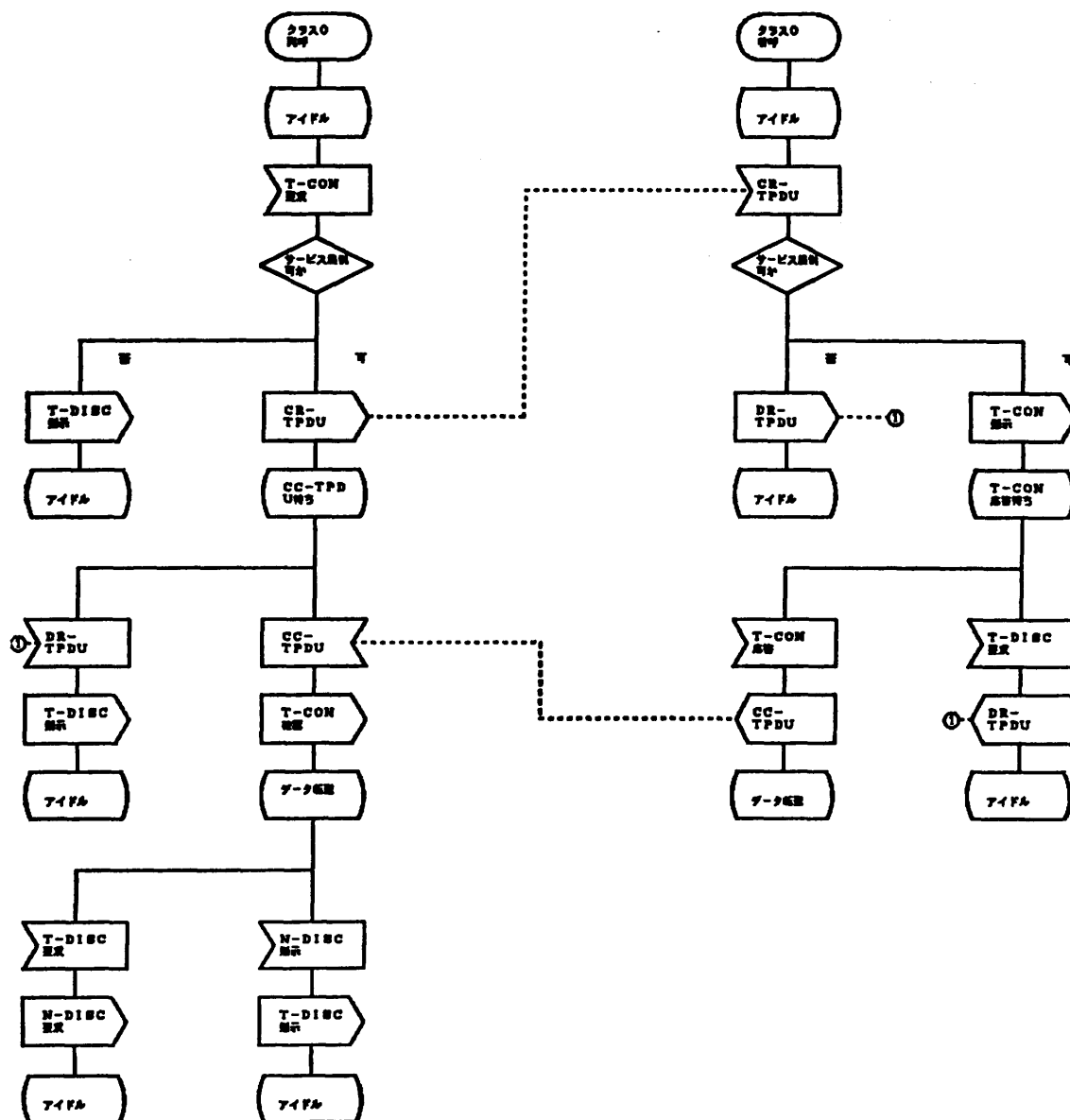


図 7 同期記述の例

Fig. 7 An example of synchronous processing description.

示し、処理の流れを追いやすくしている。関連ある手続き図の呼出しは、マウスで手続き呼出しシンボルをクリックすると手続き図に関する情報をもったファイルをアクセスし、手続き呼出しシンボルと同じ名前をもった手続き図を選択することによって実現した。

このように作成された画面は合成された図の名称および表示位置、同期記述の場合はさらに信号線の情報をファイルに保持する。再使用の場合は単独で作成された各プロセスグラフにこのファイルの情報を付加して関連するプロセスグラフの同時記述を実現した。こ

のため個々の図を後に修正しても、その結果が同時記述の図に反映できる特徴をもっている。図 7 に同期記述の例を示す。この図は図 3 で示したトランスポートプロトコルのクラス 0 の着呼側（右側）と、それに対応する発呼側（左側）との同期関係を示している。破線は信号線を表している。

4.4 出力フェーズ

紙上でも信号の流れや処理の流れを検査するために印刷機能を支援している。用紙の大きさは A3 (縦長), A4 (縦長, 横長), B4 (縦長, 横長) が選択でき、図

が大きくなる場合は複数枚に分割可能とした。

5. 現状と評価

【全体】

本エディタはC言語を主に用い、一部 FORTRAN 77 言語で記述されており、コメントを含め全体で約 70 K ステップのソフトウェアとなった。

エディタにおいては応答時間の早さが使いやすさに大きく影響する。本エディタでは一つのシンボルを画面上に挿入する時間を約 1.8 秒で実現した。また、SDL/PR から SDL/GR への変換はシンボルが 20 個程度の場合約 30 秒かかり、また SDL/GR から SDL/PR へも同様な時間がかかるが、いずれも許容範囲内で応答することが分かった。ただ、応答待ちしている時間はソフトウェアが動いていることを示す何らかのマークが必要である。

【入力フェーズ】

情報通信システムの設計者が数十シンボル程度の図を下書きが既にある状態で完成させる場合、シンボルの位置の指定やフロー線の接続、文字の記述等でおおよそ 1 シンボル 1 分の時間が必要であった。これは手書きに比べて時間がかかるが、修正等の編集フェーズのことを考慮に入れば十分実用的な数字であった。

SDL/GR 補助入力方法では入力時に既にシンボルの種類が分かっていることが多いので、この時点での各々のシンボルへの変換および分岐方法の実現が今後の課題であることが分かった。

【編集フェーズ】

従来のように切り貼りしたり、消しゴムを使って書き直す必要がなくなったため、修正が簡単に行えるようになったのが大きなメリットであった。

CCITT 勧告準拠を設計基準としたため、文字はシンボル内にはみださないように入れ、余りの部分はコメント部に入れた。現在、シンボル内には 12 文字 (6 文字×2 行)、コメント部には 48 文字入るので画面上では文字数は十分である。さらに、印刷時に見やすくするためにシンボル内の文字数を増やすことを検討したが、判断シンボルにおいては、はみだしを許すか、もしくは他のシンボルより大きくする必要があることが分かった。また現在、分岐線は判断シンボルの下から出ているが、横から出る方が自然であることが分かった。

また、視覚的に全体のモジュール構成がマルチウィンドウによって表示されているので全体の構成を念頭

において個々のプロセスグラフを編集できることが分かった。なお、1 画面に表示されるシンボル数は現在最大 8×22 個であるが、現在の使用程度ではまだ問題は生じていない。

【検査フェーズ】

記述上の誤りの検査は、SDL/GR から SDL/PR への変換ができることを前提としているため、仕様書の記述に使用するためには厳しすぎる場合があった。例えば、プロセス図を作る時は全体の一部分だけを画面に作るので遷移する先の状態シンボルだけが存在し、それ以下の動作の記述を別の画面に記述する場合が多い。これは厳密に言えば誤りであるが、これを誤りとしないう制限を緩くする必要がある。

信号の流れを追うため最大 5 つのプロセスグラフまで同時に 1 画面に表示できるが、現状では異なるプロセスグラフ間の信号の流れを確認するためや、手続き呼出しの内容の確認のために 2 つのプロセスグラフを記述することが主であることが分かった。

【印刷フェーズ】

当然のことではあるが、手書きの場合と比較して格段に美しく、見やすいことが分かった。

印刷用紙では縦方向に最大の A 3 (縦長) と横方向に最大の B 4 (横長) の使用が主であり、できる限り 1 枚の用紙に記述する傾向があることが分かった。また、A 4 一枚に縮小印刷して文字が読めなくてもフローが分かれば作図の過程では十分である場合も多いということが分かった。

6. む す び

SDL グラフィックエディタをワークステーション上に実装した結果、日本語仕様書等ドキュメントが、入力、編集、検査および印刷の各フェーズを総合すると、手書きで記述した場合と遜色ない時間でしかも見やすく作成できることが分かった。今後は、操作性を損なわない範囲でいかに CCITT 勧告を完全に遵守するか、また、SDL で記述した内容をもとに、いかに通信ソフトウェアの自動生成や自動検証を行っていくかが課題である。また、SDL は情報通信システムだけではなく、マンマシンインタフェースの記述やオフィスにおける伝票処理等への適用も可能であり^{15), 16)}、今後、さらにその応用形態について検討を加えていきたい。

謝辞 日頃、御討論いただく三菱電機(株)情報電子研究所の諸兄に深く感謝する。

参 考 文 献

- 1) 白鳥, 高橋, 野口: NESDEL: プロトコル向き仕様記述言語とその応用, 情報処理学会論文誌, Vol. 26, No. 6, pp. 1136-1144 (1985).
- 2) v. Bochmann, G.: Usage of Protocol Development Tools: The Result of a Survey; Protocol Specification, Testing and Verification VII An International Symposium, pp. 147-170 (1987).
- 3) v. Bochmann, G. (水野, 井手口(訳)): 分散処理システム・デザイン, 工学社, 東京 (1987).
- 4) ISO: Information Processing Systems—Open Systems Interconnection—Estelle—A Formal Description Technique Based on an Extended State Transition Model. ISO/DIS 9074 (1987).
- 5) ISO: Information Processing Systems—Open Systems Interconnection—LOTOS—A Formal Description Technique Based on the Temporal Ordering of Observational Behavior. ISO/DIS 8807 (1987).
- 6) CCITT: Functional Specification and Description Language (SDL). Recommendations Z. 100-Z. 104 (1985).
- 7) 水野忠則: プロトコルの形式記述とコンFORMANCE試験, 情報処理, Vol. 26, No. 4, pp. 420-427 (1985).
- 8) 伊藤, 市川: 並行プロセスを基本とした交換プログラム仕様の階層的検証, 電子通信学会論文誌, Vol. J 69-B, No. 5, pp. 449-459 (1986).
- 9) 大平, 武居, 新保: ISDN に関する一考察ユーザ/網インタフェースにおける端末整合性に関する一検討, 電子通信学会技術研究報告, SE 86-62, pp. 19-24 (1986).
- 10) 乗越雅光: グラフィックディスプレイを用いたSDL チャートの作成, 管理支援ツール, 電子通信学会技術研究報告資料, SE-82-106, pp. 9-15 (1983).
- 11) 渡辺, 片岡, 三井: 三菱 ME 1000 シリーズ, bit 別冊高機能ワークステーション, pp. 166-175 (1987).
- 12) 高野, 水野, 坂下, 北畠: 分散型開発システムで仕様書作成からプログラム生成までを支援する, 日経エレクトロニクス, No. 425, pp. 179-194 (1987).
- 13) CCITT: Data Communication Networks: Open Systems Interconnection (OSI), System Description Techniques, Recommendations X. 224 (1985).
- 14) 原田賢一(編): 構造エディタ, 共立出版, 東京 (1987).
- 15) Hornbach, B. H.: MML: CCITT Man-Machine Language, *IEEE Trans. Commun.*, Vol. COM-30, No. 6, pp. 1329-1336 (1982).
- 16) 宗森, 水野: オフィスの形式的記述へのアプローチ, OA 学会誌, Vol. 6, No. 3, pp. 59-60 (1985).

(昭和 62 年 4 月 22 日受付)

(昭和 63 年 4 月 14 日採録)



宗森 純 (正会員)

昭和 30 年生. 昭和 54 年名古屋工業大学電気工学科卒業. 昭和 56 年同大学院工学研究科電気工学専攻修士課程修了. 昭和 59 年東北大学大学院工学研究科電気及通信工学専攻博士課程修了. 工学博士. 同年三菱電機(株)入社. 現在同社情報電子研究所システム・ソフトウェア開発部所属. システムアーキテクチャ担当. 脳の情報処理機構, 特に視覚系の情報処理機構に興味をもっている. 電子情報通信学会, オフィス・オートメーション学会各会員.



水野 忠則 (正会員)

昭和 20 年生. 昭和 43 年名古屋工業大学経営工学科卒業. 同年三菱電機(株)入社. 現在同社情報電子研究所システム・ソフトウェア開発部. 工学博士. 情報通信システムおよび分散処理システムに関する研究・開発に従事. 著書としては『マイコンローカルネットワーク』(産報出版), 『分散処理システムデザイン』(共訳, 工学社), 『電子メールとメッセージ通信』(監訳, 工学社)などがある. 電子情報通信学会, オフィスオートメーション学会, IEEE 各会員.