

## 対話方式設計に関する一考察

松谷 泰行      岡野 寿夫  
多摩大          筑波技術短大

ここで論ずることは、ビジネスアプリケーションに於ける対話式の入出力設計に関する方法論である。システムの分析・設計の過程で、対話方式の設計と、そこで得られるデータに基づく要求仕様の設定とを、同一に論じようとすることから、例えばデータフローダイアグラムが複雑化する等、混乱を招くことがしばしば見受けられる。ここでは、状態遷移図を応用した入出力ダイアログを作成し、それをもとにデータフローダイアグラムを展開することにより、対話部分を独立に設計し、かつ本来のシステムと整合性が確保される方法論を提案する。

### A DISCUSSION ON INTERACTIVE INPUT/OUTPUT DESIGN

Hiroyuki Matsutani

Toshio Okano

Tama Institute of Management  
and Information Sciences

Tsukuba College of Technology

4-1-1, Hijirigaoka, Tama-shi,  
Tokyo, 206 Japan

4-3, Amakubo, Tsukuba-shi,  
Ibaragi, 300-26 Japan

We will discuss about a method for interactive input/output design in business application areas. Interactive input/output design and requirement specification based on the data obtained in the input/output design are often performed simultaneously during system analysis/design process, which result in a considerable confusion with complicated data flow diagrams etc. We will propose a method which utilizes data flow diagrams based on the input/output dialog developed from status transition diagrams, independently performs design of interactive part and perfectly coincides with original system functions.

## 1 問題の所在

ビジネスアプリケーションに於いて、データの入出力をエンドユーザーが対話的に端末から行うことは、最早常識といってよい。ここで発生する設計上の問題の一つを取上げ、それについて論ずることとしたい。以下、簡略化した例題を挙げ、それを参照しつつ議論を行うこととする。なお、分析手法としては DeMarco<sup>[1]</sup> に準拠して記述する。

### 〔例題 A会社 受注処理業務〕

- (1) A社は卸問屋であり、受注処理業務をオンラインリアルタイムで処理するシステムを設計中である。
- (2) A社の受注処理システムは次の様な考え方で検討されている。
  - ① 顧客との取引は、取引条件に関する基本契約が締結されてからしか、実際の取引を行わない。  
(新規の顧客がいきなり物を発注してくることはない)
  - ② 基本契約締結後、顧客は電話で発注してくる。  
倉庫に品物があり希望納期に間に合う時は直ちに出荷の手続きを取る。  
上記の条件に合わない時、発注は行われぬ。なお倉庫は数箇所があり品物はどこにあってもよいものとする。
- (3) 以上の②部分は、図 1.1に示すデータフローダイアグラム<sup>[1]</sup> (以下DFDと呼ぶ)の形でマン・マシン分担が考えられている。但し下記の事項が前提である。
  - ① 受注担当者は顧客地域別・品種別に担当範囲が決まっており、顧客コード品名コード等を熟知している(速やかにコードで入力可能)。
  - ② さはさりながら、入力ミスの発生は予測され、コードのチェック、顧客名品名等の表示による確認等が要求されている。
  - ③ 注文は、届け先別・納期別に複数の品物が発注される。1つでも品物が納入できない時、その注文は受注できない。注文1件(届け先別・納期別)毎の品物の種類はせいぜい数件である(1画面に収まる)。

図 1.1に示すDFDでは、受注担当者は「注文内容」を入力し、「受注可否」を受け取り、顧客にその旨を返答する。ここには入力ミスの修正・確認等、幾つかの対話が発生しそれに基づくデータフローが発生する。しかし、それを図 1.1の全体を俯瞰するDFDに記すことは必ずしも得策ではない。それはDFDを複雑化し、何が本質的な入出力かを見失うおそれがある。また対話のやり方自身、ある意味で楽しい議論であるとともに、なかなか収容のつかない一面もあり、そこにベテラン設計者がひきずり込まれると、他の重要事項(受注可否のロジック等)への検討がおろそかになる危険もある。上記の問題意識から、入出力部分を切り離して検討する進め方を議論することとしたい。

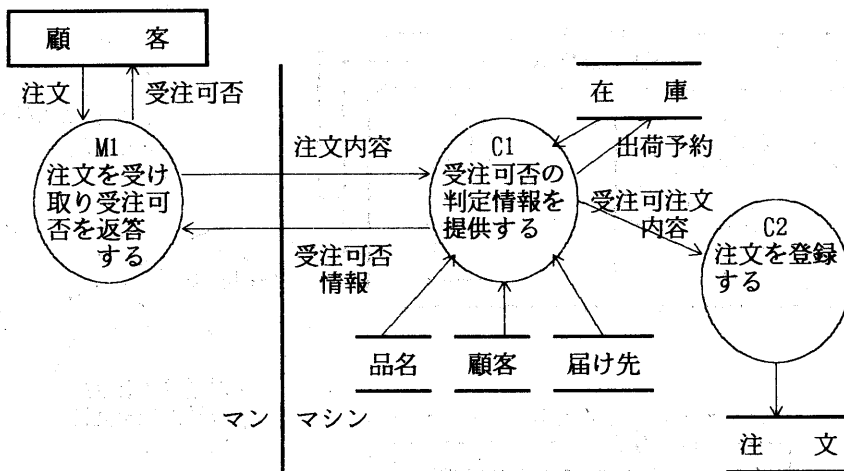


図 1.1 A社 受注処理システムDFD

## 2 対話式入出力設計方式

対話式の入出力設計をいわゆる本体部分と切り離して行うには、次の条件を満足する必要がある。

- (1) 対話式入出力の範囲を明確にすること
- (2) 対話部分をコンピュータ処理をあまり考慮せず設計する設計技法を考案すること  
(コンピュータをよく知らないエンドユーザーでも書ける表記法)
- (3) 対話部分が全体のプログラム構成のどこに位置づけられるかを明確に示せること
- (4) しかして、技術的なチェックが適宜行われるプロセスを考えること

上記を考慮した設計方式が図 2.1に示すものである。すなわち次のようになる。

- (1) まず、対話式の入出力部分を切り出し、対象範囲を明確にする。具体的な表記法としてはDFDを用い、マン・マシン境界上にある機能（バブル）を構成することにより対話部分であることを明らかにするとともに、(4)で検討されるDFDがこの機能の下位部分である位置づけを明確にする。
- (2) 上記で規定された範囲の対話方式を状態遷移図を応用したダイアログ記法により表現し、同時に画面設計等を行う。これらを入出力ダイアログと呼ぶ。
- (3) 入出力ダイアログが技術的に実現可能かをチェックし、問題あれば修正する。
- (4) 以上で出来た入出力ダイアログをDFDに展開する。このDFDから、コンピュータ部分については、構造化設計<sup>[2]</sup>の見地より、いわゆる本体部分との接点が明確化され、マン部分については業務マニュアル等作成の元資料となる。
- (5) 本体部分と合流し全体としての設計がまとまる。

なお、ここではキーボード・ディスプレイ端末を使った対話をモデルに検討する。

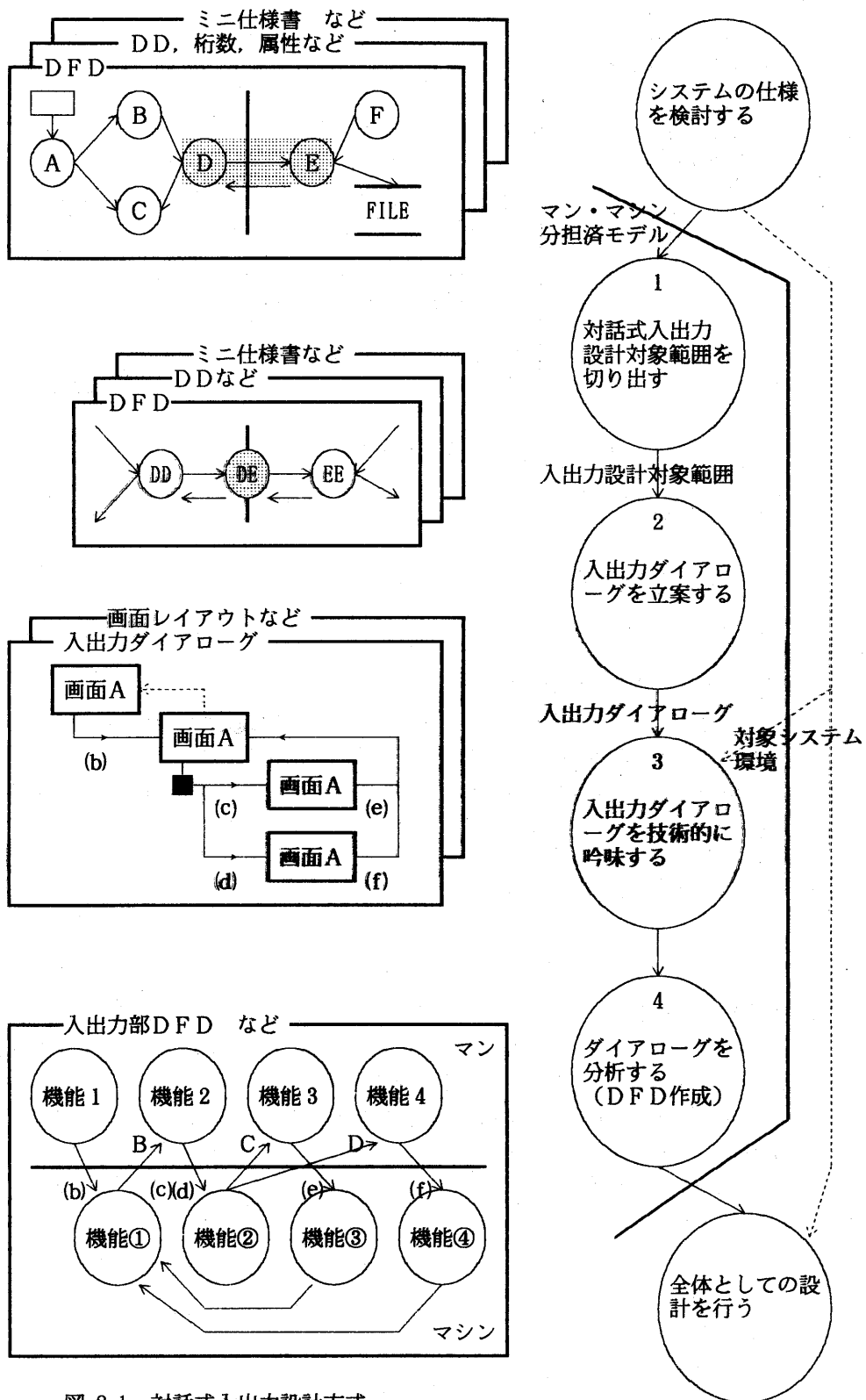


図 2.1 対話式入出力設計方式

### 3 対話式入出力設計対象範囲の切り出し

対話式入出力設計対象範囲は次のような考え方で切り出す。

- (1) 切り出しの対象は、マン・マシン境界を横切るデータフローである。
- (2) 切り出しに当たっては、入出力作業として1つの作業単位を形成する範囲毎に切り出す。ここに作業単位というのは、連続して行われる作業の集まりを言う。例えば注文を入力し、注文請け書を発行する業務を考えた時
  - ① 1件の注文入力に対し、直ちに注文請け書が発行される場合は、それで1つの作業単位が形成されるが(図 3.1)
  - ② 注文入力が終わってから、改めて注文請け書がまとめて発行される場合は2つの作業単位に分かれる(図 3.2)

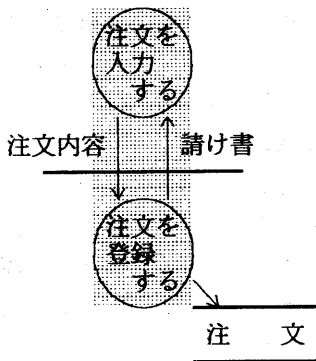


図 3.1 1つの作業単位

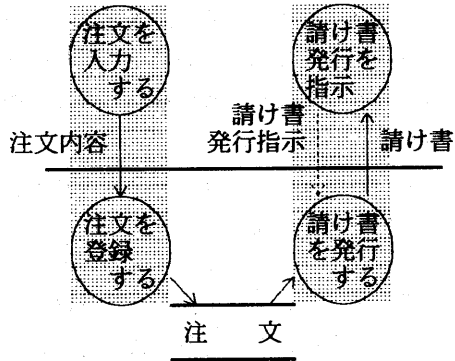


図 3.2 2つの作業単位

- (2) マン・マシン境界の両側の機能(バブル)の中には、入出力以外の本来機能に関するものが含まれている場合がある。これは切離し入出力に関する機能のみ(例えばチェック機能)を切り出す。例題の図 1.1の場合、機能 C1 にある在庫からの受注可否判定機能を除き切り出されることになる(図 3.3)。

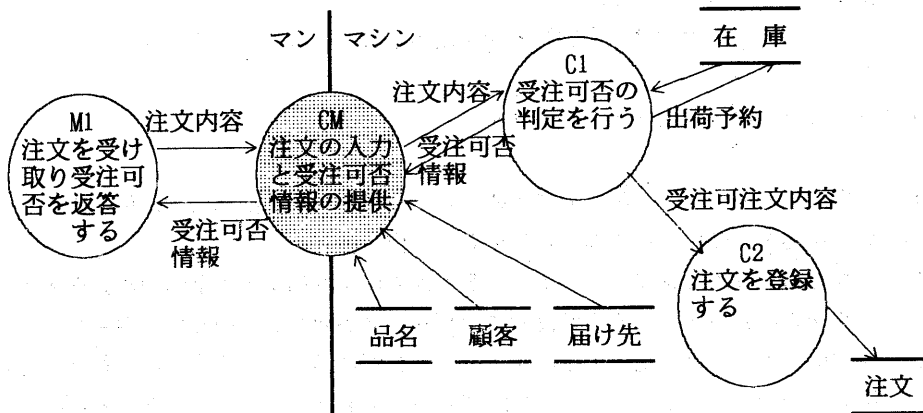


図 3.3 入出力対象機能の限定(例題の場合)

#### 4 対話のデザイン：入出力ダイアログの作成

どのように対話して入出力を行うかのデザインは、入出力ダイアログで表現する。入出力ダイアログとは、対話の状況を図式に表現したものであり、状態遷移図の考え方を応用している。例えば図 3.3の例で、入出力の対話を、

- (1) 仕事を始めようとして端末に向かった時は、システムの初期状態にあるものとし、「注文の入力と受注可否情報の提供」のプログラムを起動させると、入力開始画面(図 4.2)が表示される。
- (2) これに対して、顧客コード、納品日、届け先コード、品名コード、数量等を入力する。注文1件分の入力が終わると(1件終了を示す何らかのキーが押されると)、入力された結果はチェックされ、エラーがあれば入力エラー画面(図 4.3)が表示され、修正入力を行うこととなる。この修正入力も再度チェックされ、エラーがあると再度入力エラー画面が表示され、以下エラーがなくなるまで繰り返される。
- (3) エラーがない入力については(エラー画面から正しく修正されたものも含め)、顧客コード等を顧客名等に変換した入力確認画面(図 4.4)を表示し、入力結果の確認を行う。確認は次の3つのいずれかの形を取る。  
1 : 内容OK    2 : ミスあり修正    3 : 思い違い等あり始めからやり直し
- (4) “内容OK”の場合は、在庫状況が検討され結果として受注可否画面(図 4.5)が表示される。この受注可否画面には受注可否判断とともに“注文入力を続けますか Y/N”のメッセージが出、“Y”だと継続入力画面が表示され、“Y”でなければ終了ということでシステム初期状態に戻る。
- (5) 継続入力画面(図 4.6)は、直前の入力内容が表示されており、変更箇所を修正する形で入力する。入力結果はチェックされ、エラーだと入力エラー画面(図 4.3)が表示される。エラーでない場合は入力確認画面(図 4.4)が表示される。
- (6) “ミスあり修正”の場合は、入力修正画面が表示される。入力修正画面(図 4.7)は、直前の入力内容が表示されており、修正箇所を修正する。入力結果はチェックされ、エラーだと入力エラー画面(図 4.3)が表示され、エラーでない場合は入力確認画面(図 4.4)が表示される。
- (7) “やり直し”の場合は入力開始画面が表示され、これに対し入力する。

とすると、図 4.1の入出力ダイアログを描くことになる。

現実の作成に於いては次の考え方で進める。ダイアログ記法は表 4.1に記す。

- (1) 自分が端末の前に座って作業する積もりで作る。
- (2) すなわち、「画面を見て入力する、その入力により何が出て欲しいか」を考え、それを素直に表現する。その背後でコンピュータが何をしているかは考えない。
- (3) 作成に当たっては上記のようなストーリーを頭に描きながら作る。それとともに、個々の画面レイアウト(スケッチ的でも可)を描き、ダイアログと画面レイアウトそして画面に対する説明を以てワンセットとする。

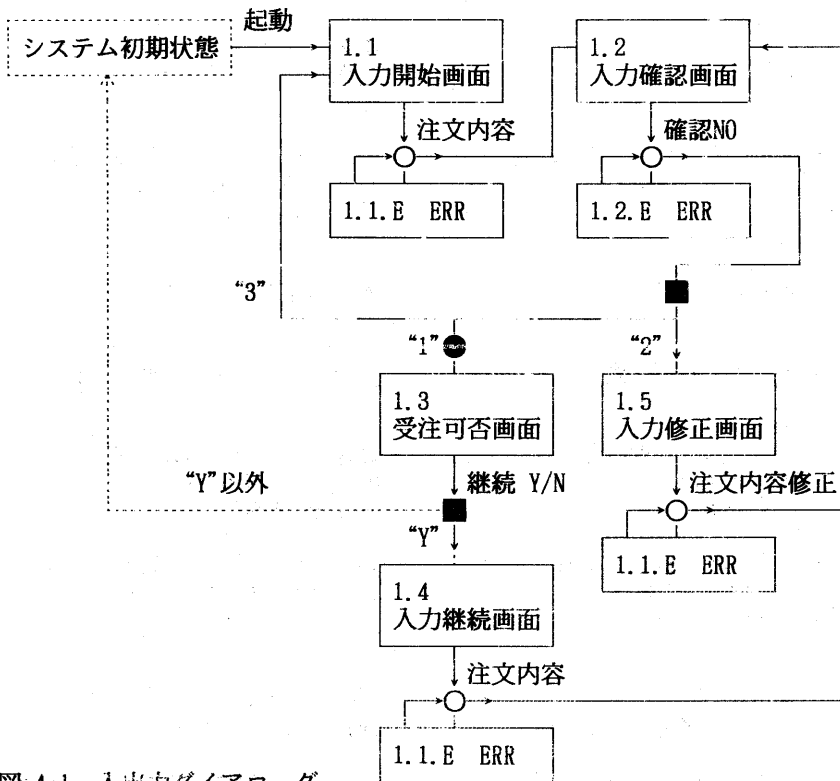


図 4.1 入出力ダイアログ

表 4.1 ダイアログの表記法

 	<p>画面AにデータXを入力すると画面Bになる。</p> <p>画面は四角で表す。画面NO、画面名を記す。 点線は終了の時の流れ。</p>
	<p>データXが “1”の時 画面B “2”の時 画面C を表示する。</p> <p>データの値により異なる画面を出す時</p>
	<p>データXに対するエラーチェック 左 OK, 下 NO</p>
	<p>入力Xで入力としての目的を達成 (●の時点)</p>
	<p>階層的表現 (図 4.1には例なし) Sには多くの画面とその遷移がある。 画面が多くて一枚に書ききれないような時に用いる。</p>

顧客 XXXX  
 納期 XX年XX月XX日  
 届け先 XXXX

NO	コード	品名	数量	在庫	可否
01	<u>XXXX</u>		<u>XXXX</u>		
02	<u>XXXX</u>		<u>XXXX</u>		
.					
.					

データを入力して下さい

図 4.2 入力開始画面 (画面 1.1)

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1989 ERR

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>		
02	<u>0903</u>	ERR	<u>0200</u>		
.					
.					

入力内容を修正して下さい

図 4.3 入力エラー画面 (画面 1.1.E)

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>		
02	<u>0803</u>	フロッピー	<u>0200</u>		
.					
.					

確認 X (1:OK 2:修正 3:やり直し)

図 4.4 入力確認画面 (画面 1.2)

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>		
02	<u>0803</u>	フロッピー	<u>0200</u>		
.					
.					

1~3 以外入力 もう一度  
 確認 4 (1:OK 2:修正 3:やり直し)

図 4.4-1 入力確認エラー画面 (画面 1.2.E)

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>	0500	OK
02	<u>0803</u>	フロッピー	<u>0200</u>	1000	OK
.					
.					

受注可 出荷予約・注文登録完了  
 注文入力を続けまか Y/N X

図 4.5-1 受注可否画面 (画面 1.3) 例 1

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>	0500	OK
02	<u>0802</u>	ポーター	<u>0150</u>	0100	NO
.					
.					

受注不可  
 注文入力を続けまか Y/N X

図 4.5-2 受注可否画面 (画面 1.3) 例 2

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>	0500	OK
02	<u>0803</u>	フロッピー	<u>0200</u>	1000	OK
.					
.					

データを入力して下さい

図 4.6 入力継続画面 (画面 1.3) 例 1

顧客 0505 情報処理商会  
 納期 90年05月15日  
 届け先 1990 本店

NO	コード	品名	数量	在庫	可否
01	<u>0101</u>	WPプリン	<u>0100</u>	0500	OK
02	<u>0803</u>	フロッピー	<u>0200</u>	1000	OK
.					
.					

データを修正して下さい

図 4.7 入力修正画面 (画面 1.3) 例 2



## 5 入出力ダイアログの技術的吟味

入出力ダイアログは、コンピュータをよく知らないエンドユーザーでも設計出来ることを目的に考案した。現実にエンドユーザーが設計した場合には、斬新な方式が期待出来る一方、使用するシステム環境（ハード・OS・DCソフト等）から見て実現不可能なものを考えるおそれがある。その観点から技術的な実現可能性（やりやすさ、やりにくさを含め）の吟味を行う。具体的な内容はシステム環境と密接に関係すること故、一般的に論ずることは難しいが、例えば次のような項目が吟味の課題となる。

- (1) コンピュータと端末間のデータ転送方式、例えば画面単位の転送が容易か、項目単位の転送の方がやりやすいか
- (2) 画面レイアウトの実現性（桁数、属性等）
- (3) ファンクションキー等の使い方に関する問題 など

## 6 ダイアログの分析：DFDによるプログラム設計課題の吟味

実現可能と判断される入出力ダイアログについては、プログラム設計につなげる意味での課題を検討し、可能な限り本体部分と並行してその設計を行う。ここではDFDを用い主として本体部分との整合性を検討する。

DFDは、バブルで表現される機能にデータが入り、それが出力データに変換される（ 6.1）。一方ダイアログはデータをすることで画面が出力され（ 6.2）背後にコンピュータ機能が存在する。このことから 6.2のダイアログは 6.3のDFDで表現されることになる。



図 6.1 DFD

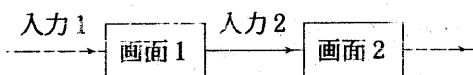


図 6.2 入出力ダイアログ

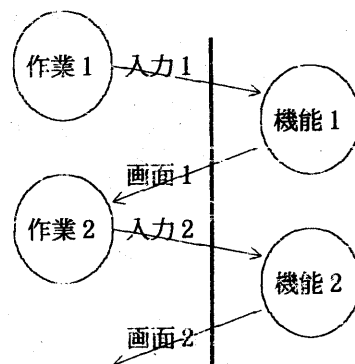


図 6.3 ダイアログからDFDへ

以上の関係より、 4.1の入出力ダイアログをDFDに変換したのが 6.4である。見方としてのポイントを述べると次のようになる。

- (1) ハッチングを施した機能 CM.06が本来機能の部分である。これは目的達成（●）から導かれる。
- (2) CM.M2, CM.M5, CM.M6 では修正箇所のみするが、コンピュータには注文内容全部が転送されるものと考えている。
- (3) 点線はデータの流ではなく、制御の流れを意味する。

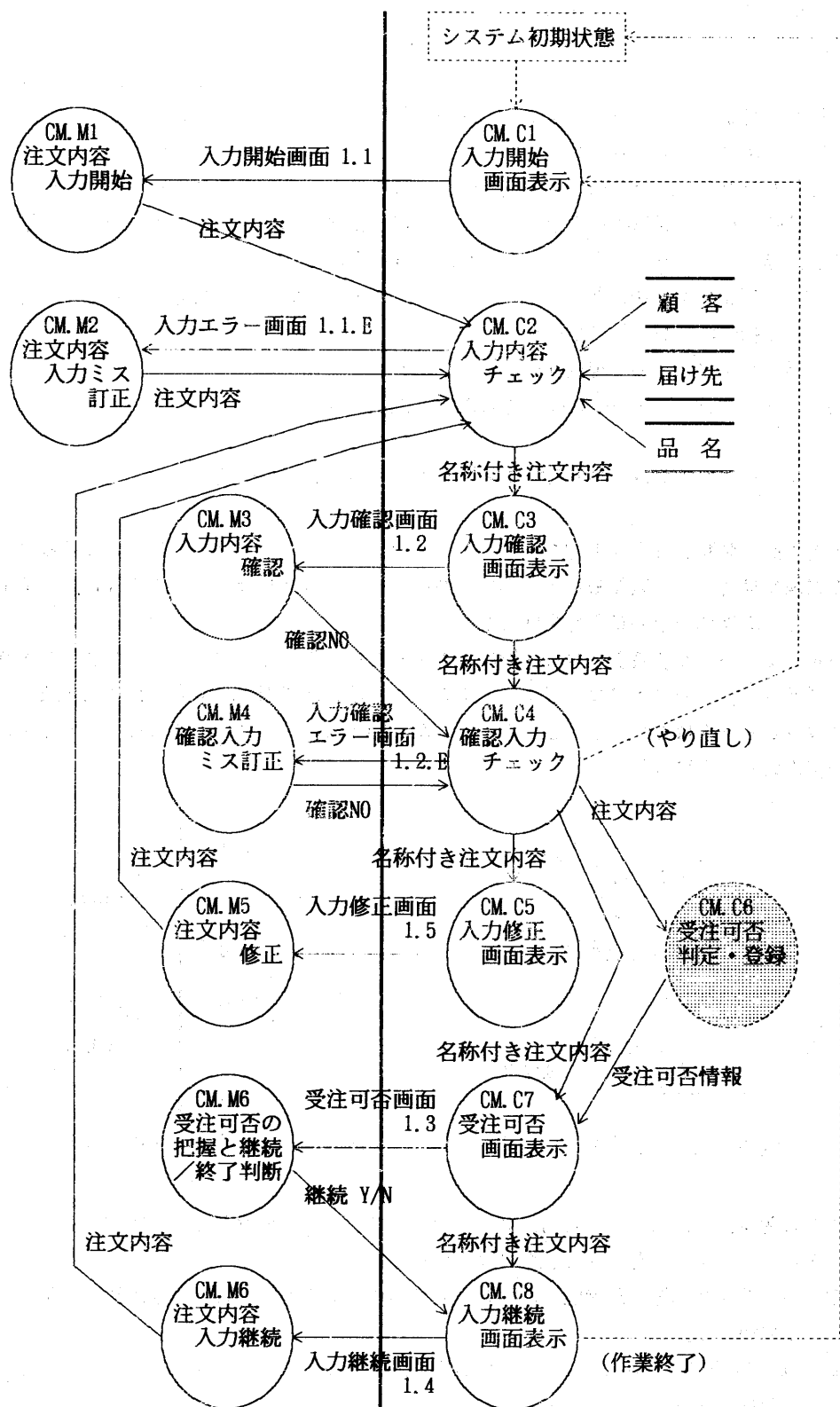


図 6.4 入出力ダイアログ例のDFD

このDFDを用いてプログラム設計上の問題を幾つか議論できるが、ここでは紙数の関係からモジュール構造の観点について述べる。構造化設計に於ける、入力（源泉）→変換→出力（吸収）の変換分析を用いて図 6.4を整理すると図 6.5が得られ、この構造はモジュール構造として図 6.6となるのが予想される。すなわちDFDで表現することにより、全体としての整合性確保の検討が容易であり、入出力部分の独立性を保証してゆく方策をたてることが出来、入出力部分の設計を並行して行うことの技術的な裏付けになる。

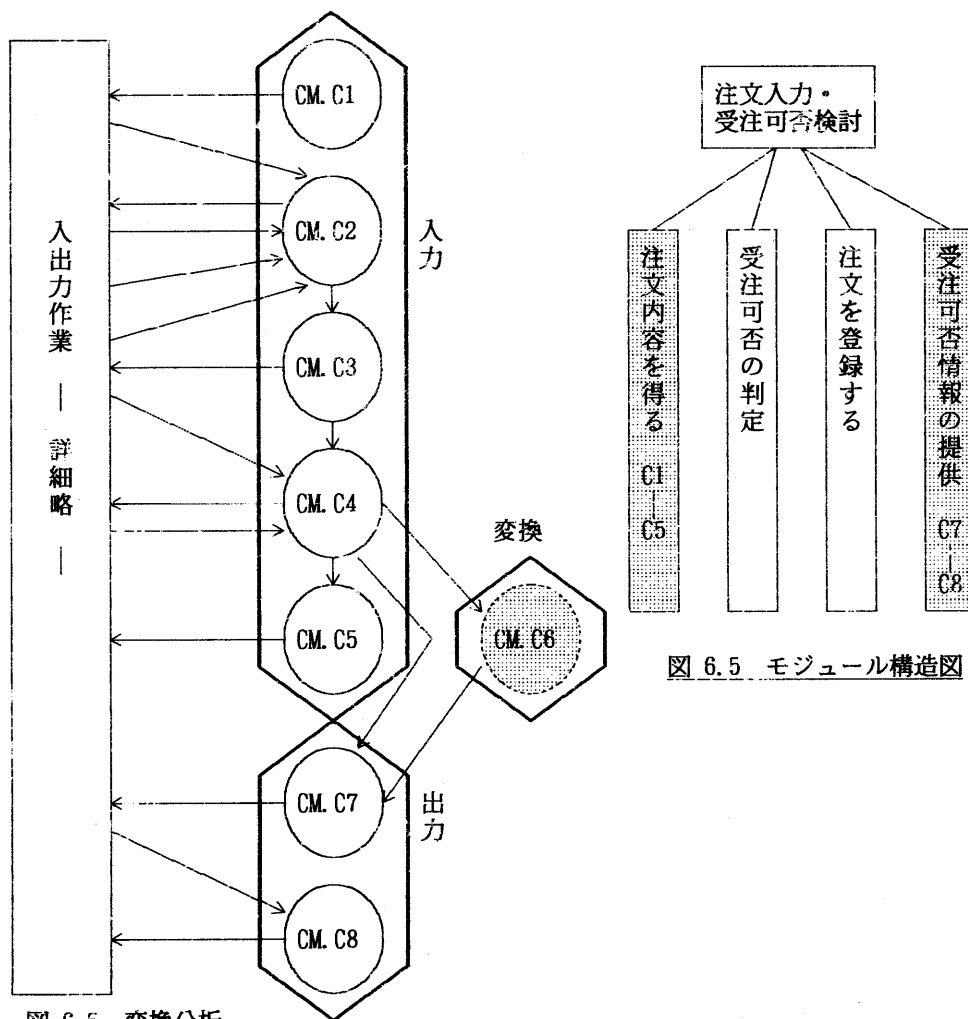


図 6.5 モジュール構造図

図 6.5 変換分析

## 7 終わりに

以上の議論は、まだ試案段階であり細部の検証は不十分である。ただ入出力ダイアログの考案によるエンドユーザーの参画、DFDによる設計の並行化で、実現の可能性はあると確信している。多くの方の御意見を期待するところである。最後に、有益な意見交換を行った新口鉄情報通信システム(株)の方々に深く感謝の意を捧げる次第である。

### [参考文献]

- [1] Tom Demarco, Structured Analysis and System Specification, 1978
- [2] Meilir Page-Jones, The Practical Guide to Structured Systems Design, 1988