

事象トレース付きデータフロー図を用いた システム設計手法についての一考察。

葛山善基 神代知範
滋賀大学経済学部情報管理学科

従来、我々は業務処理システムの開発において事象トレース図を記述した後データフロー図を記述していた。しかし、データフロー図の記述には事象トレースの情報を含ませて記述することができず、大きなデータの流れが把握しにくいという問題があった。我々は、新しいモデルである事象トレース付きDFDを提案する。さらに、このモデルを用いたオブジェクト指向的システム設計手法を示した。この設計手法においては、(1) オブジェクトの分類、事象トレースと主なDFDの記述；(2) オブジェクト毎の事象トレースおよびDFDの分析；(3) 事象トレースと主なDFDを事象トレース付きDFDに変換；(4) 詳細なDFDの追加を行う。

A system design method using Data Flow Diagrams with Event Traces

Yoshiki Katsuyama, Tomonori Kumashiro
Department of Information Processing and Management
Faculty of Economics
Shiga University

In the stages of system analysis, we usually describe an event trace first followed by a data flow diagram (DFD). However, a DFD does not contain event trace information, and it is difficult to extract this information from it. We propose a new model (DFDET) which contains event trace information. Furthermore, we propose an object-oriented system design method, which (i) describes the system outline, the objects, the event trace, and the main DFD; (ii) analyzes the event trace and the main DFD; (iii) translates the main DFD and event trace into a DFDET; and (iv) adds further DFD information to the DFDET. This system design method using the DFDET makes a DFD easy to understand and allows a systems analyst to analyze systems easily.

1. はじめに

システム分析の記述法としては、1971年にデマルコが提唱した(1)データフロー図(DFD)が現在でも用いられている(2)。このDFDは対象としている業務内でのデータの流れを記述しているものであり、対象としている業務と他の業務(システム)とのデータのやり取りは通信等で利用されている事象トレース図を用いて記述されている(3)。

従来、このDFDは個々のプロセス間でのデータのやり取りのみ記述し、データの一連の流れを読みとることが難しかった。また、処理の手順はプロセスの中でのみ記述され、データフローの大きな流れが理解しにくいものであった。また、事象トレースはシステムと他のデータ源泉・吸収とのオブジェクトのやり取りを記述するものであった。

従来の設計手順はDFDと事象トレースの記述が独立して記述され、それぞれの記述も思いつくまま記述していくことが多かった。それゆえ、DFDの実行順序を把握することは困難であった。また、DFDのプロセス間やプロセスとデータベースとの間のデータのやり取りを事象トレースのように記述するユースケース(4)は並行して実行してもよい部分についても直列に記述するため、実際の仕様を変えるものである。

本稿では、DFDのデータ源泉・吸収とプロセスとのやり取りを事象トレースとしてとらえ、この事象トレースをDFDの左右に記述できる事象トレース付きDFDを提案する。従来、事象トレースにおける繰り返し処理は状態遷移図を別に記述して表したが、事象トレース付きDFDにおいては、事象トレースの所に繰り返し制御が記述できようにし、他の情報を見なくても繰り返しが分かるようにした。それゆえ、DFDETは新しいモデルである。

さらに、ビジネスプロセスのオブジェクトに着目した事象トレース付きDFDの段階的な設計手法を示す。この段階的な設計手法がいかに有効であるかをデマルコの事例を用いて実証する。

2. 従来のデータフロー・ダイアグラムと事象トレース

2.1 データフロー・ダイアグラムの定義

データフロー・ダイアグラム(DFD)は、データフロー、プロセス、ファイル、データ源泉・吸収の4つの基本要素から構成されている。図1に示すDFDは上記の4つの基本要素をすべて含んでいる。このDFDの動作は以下になる。

データ源泉「海運会社」から出た「積荷」はプロセス「コンベアーシステム」により、ファイル「倉庫」に格納される。



図1 データフロー図の例

2.2 事象トレース

事象は時間軸上のある点で発生する出来事のことである。一般的には、この出来事が情報として他に送られ知らされる。このため、あるものから他のものへデータが送られた時、それを事象と考え、外部のデータ源泉・吸収とのデータのやり取りを時間的に記述したものが事象トレースと考えることができる。対象システムや外部のデータ源泉・吸収は垂直線として表され、矢印付きの水平線が事象を表している。時間は上から下へと流れるが、その間隔には意味を持たせていない。それは事象の順

序のみ表している。また、事象トレースはやり取りの一例にすぎず、従来は実現手段である状態遷移図を付けていた。

通関処理における対象システムと外部のデータ源泉・吸収とのやり取りの一例を事象トレースとして図2に記述する

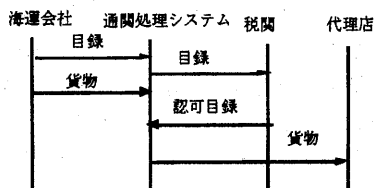


図2 通関処理システムと外部データ源泉・吸収間との事象トレース

3. ビジネスプロセス分野のための新しいモデル

3.1 事象トレース付きDFD

データ源泉・吸収とプロセスとのデータのやり取りは、データ源泉・吸収とこのプロセスを含んだシステムの間の事象と見なすことができる。それで、外部のデータ源泉・吸収を従来のようにDFDの内部に記述せずに、記述対象のシステムのDFDとは分離してDFDの左右に配置することで、DFDの左右に事象トレースが記述できる図を考案した。この記述法を事象トレース付きDFDと呼ぶことにする。従来の事象トレース図においては、対象のシステムおよびそれぞれの外部のデータ源泉・吸収を示す縦の直線の間の水平線によって事象を表現していた。しかし我々の提案した事象トレース付きDFDにおいては、システムの境界を2本の縦の直線で表し、左、中央、右の3つの領域に分割している。システムと外部のデータ源泉・吸収の間のデータのやり取りは境界線を横切る矢印で表す。途中のプロセスとファイルはすべて中央の領域に記述される。

3.2 事象トレース付きDFDの特徴

DFDでは繰り返し処理はループさせすことにより記述できるが、事象トレースはそのような記述法がなかった。事象トレースとDFDとを組み合わせるにあたり、事象トレース付きDFDでは外部のデータ源泉・吸収との境界にの繰り返し実行の記述を可能とした。事象トレース付きDFDでは、DFDと事象が同じ図の中に記述されるため、プロセスと事象との因果関係が分かる。

以下に、通関処理システムを事象トレース付きDFDで記述した。

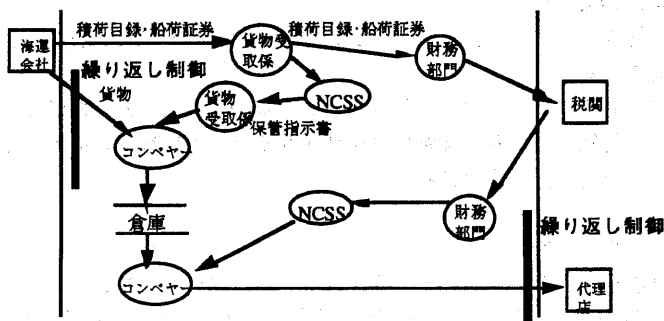


図3 通関処理システムの例

4. 業務処理システムの分析手法

業務処理システム分析においては、システム概要の記述、オブジェクトの洗い出し、事象トレースの記述、オブジェクト毎のDFDの記述を行う。

従来の業務処理システムの分析においては、オブジェクトの流れとその流れを制御する制御情報を区別することなく記述していた(1)(2)(3)(5)。そのため、DFDの全体を把握する事が困難であった。本稿ではオブジェクトとその制御情報を分けて、段階的に記述する。まず最初に主なオブジェクトについて事象トレース図とDFDを記述し、それらを分析する。

図4はデマルコの有名な例である。4章ではこのDFDに対応したDFDETを段階的に記述する設計手法を示す。

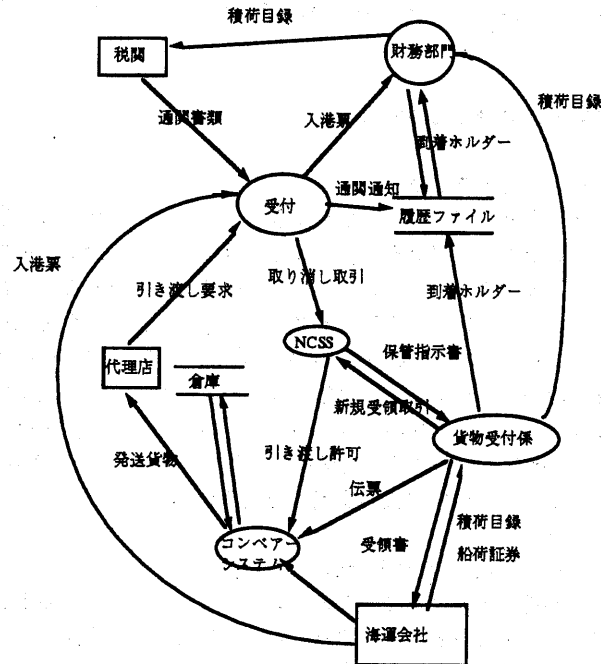


図4 通関処理システムのデータフロー図

4.1 システム概要の記述

システム概要の記述においては、このシステムが外部のデータ源泉・吸収とどのようなデータをやり取りするかを記述する。

デマルコの本の中の有名な例の一つ、通関システムの場合はシステムの概要は次の通りである：海運会社から貨物とその目録を預かり、貨物は倉庫に入れるとともに税関に検査を依頼する。税関の検査を終了した貨物については、代理店に渡す。

4.2 オブジェクトの抽出

他のデータ源泉から出たり他のデータ吸収に入るものには外部オブジェクトと制御情報がある。またシステム内だけで流れるオブジェクトを内部オブジェクトと呼び、外部オブジェクトと区別する。

(1) 外部オブジェクトの例 貨物

- (2) 制御情報の例 貨物の目録
- (3) 内部オブジェクトの例 倉庫の空き情報

4. 3 事象トレースの記述

事象トレース図には外部オブジェクトが記述されている。

図5は通関処理システムの事象トレース図である。

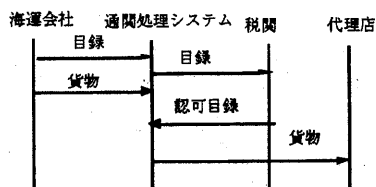


図5 通関処理システムの事象トレース図

4. 4 外部オブジェクト毎のDFDの記述

(1) 事象トレースの分析

外部オブジェクト毎のDFDを記述するには、まずすべての事象トレースを以下のようにして分析し、外部オブジェクト毎にDFDを得る。

- (i)外部オブジェクトからもっとも中心となるオブジェクトの一つを選ぶ
 - (ii)事象トレース図からこのもっとも中心となるオブジェクトを含む事象を抽出する。
 - (iii)もっとも中心となるオブジェクトの動きを記述するために、(ii)で得られた事象をつなぐ。
- その他のオブジェクトに関して(ii)(iii)を繰り返して、すべてのオブジェクトの動きを把握する。

(例) 通関システムでは貨物が中心的なオブジェクトである。まずこれについて分析をすることにする。貨物の流れている事象を抽出し、互いにつなぐ。こうして貨物が海運会社からシステムに入力されて、一時倉庫に保管され、最終的に代理店に送られることが分かる。

そのほか海運会社からシステムに送られるものに貨物の目録(情報)がある。これについて分析すると、貨物よりも前に海運会社からシステムに送られたあと税関に送られて、さらにもう一度システムに入力されていることがわかる。

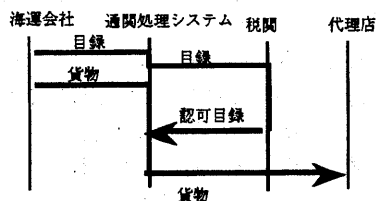


図6 外部オブジェクト毎の事象トレースの分析

(2) 外部オブジェクト毎のDFDの記述

それぞれのオブジェクトごとのDFDを以下のように記述する。

- (i)中心となるオブジェクトを選ぶ。
 - (ii)そのオブジェクトを含む事象をつないでオブジェクトのデータの流れを作る。
 - (iii)つないだ事象を表すためにDFDを記述する。ファイルをDFD中の中間点とすることがある。
 - (iv)次々にオブジェクトの流れを追って、オブジェクトに関するDFDを完成させる。
- 他のオブジェクトと制御情報に対しても(ii)(iii)(iv)の作業を繰り返して、それぞれのDFDを個別に

記述する。こうして、すべてのオブジェクトと制御情報のDFDが記述される。

(例) 税関のシステムでは貨物が中心的なオブジェクトであるので、貨物を含む事象を追っていかねばならない。貨物に関する事象トレース図によれば、貨物は海運会社から通関システムを通じて代理店へと移動する。貨物のDFDはそれらの事象をつないでいくことで作ることができる。

さらに、貨物の情報(目録)は海運会社からシステムに入力され、税関に送られて、さらにもう一度システムに入力されている。この情報についてもDFDを記述することができる。

なお、NCSSは倉庫の空きを管理し、倉庫のどこに入れるかを決定するシステムである。

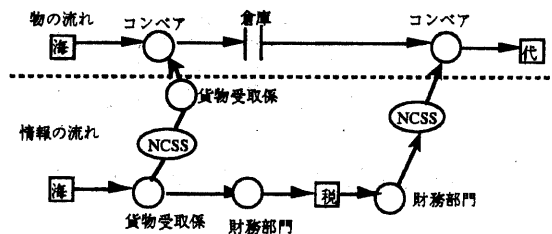


図7 オブジェクト毎のDFD

5 事象トレース付きDFDの段階的な記述

事象トレース付きDFDを記述していく手順を以下に示す。

5.1 データ源泉・吸収の配置

データ源泉・吸収を事象トレース付きDFDの左右に配置する場合に、同一側あるいは反対側に置く方がこのましい条件を以下に示す。

(a) 同じ側に配置する方が好ましい条件

(条件A) 2つの情報で処理を行う場合。

例：認可目録の中で、引渡要求のあった貨物のみ引き渡す。このような場合、認可目録と引渡要求の情報をもとに引渡の判断を行う。この2つが貨物の流れと交差することがないようにするために、税関と代理店は事象トレース付きDFDにおいて同じ側に配置する方が好ましい。

(b) 反対側に配置する方が好ましい条件

(条件B) 同一データ源泉・吸収とやり取りする事象対の2つが交互に実行する場合。

例：通関処理システムは海運会社から貨物の目録を受け取ると、この内容を税関に転送した後、倉庫に貨物を入れていく。入れられた貨物のいくつかは税関により抜き打ち検査が行われ、合格になった貨物の認可目録が税関から通関処理システム側に転送されてくる。このため、海運会社から通関処理システムへの目録転送、通関処理システムから税関への目録転送、海運会社から通関処理システムへ貨物の配送、税関から通関処理システムへ認可目録の転送はこの順番で行われる。これは、通関処理システムが海運会社とやり取りする対(目録転送、貨物の配送)と税関とやり取りする対(貨物目録、認可目録)とが交互に実行する。このため、海運会社と税関は事象トレース付きDFDにおいて反対側に配置する方が好ましい。

事象間の関連分析の必要条件を満足するようにデータ源泉・吸収を左右に配置する。

海運会社を左側に配置すると税関と代理店は右側に配置することになる。

つぎに、左右に分配されたデータ源泉・吸収を事象トレースの順番に従って配置する。税関処理した積荷が代理店に渡されるため、税関を代理店よりも上に置く。

5.2 主な流れの記述

オブジェクトごとのDFDからデータフローを転記して、外部オブジェクトの事象トレース付きDFDを記述する。次に制御情報のフローを転記する。もしデータ源泉・データ吸収とシステムのやりとりが連続的である場合には、一つにまとめて記述する。

例：まず海運会社から代理店までの貨物の流れを記述する。次に情報の流れを記述する。情報の流れは保管指示書によって貨物の流れをコントロールしたあと、税関へ送ってチェックされ、さらに戻ってきて貨物を出庫して送るための条件になっている。

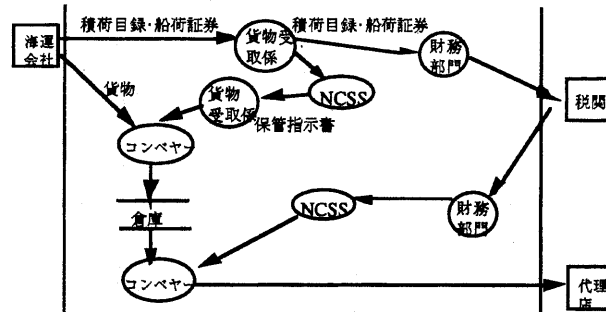


図8 外部オブジェクトのDFDET (繰り返し制御はまだ)

5.3 繰り返し制御の記述

海運会社から倉庫に積荷の移動および倉庫から代理店への移動は繰り返し行われる。これを記述すると図3になる。

6. DFDETの詳細化

5章においては、外部オブジェクトのデータフロー図が記述された。外部オブジェクトがこのDFDに従って流れるために、データ源泉・吸収からの制御情報を追加する必要がある。また、システム内の資源である内部オブジェクトの助けが必要となる。さらに、データ源泉・吸収とのやり取りは一つの窓口であるプロセスで行うため、やり取りを行っているプロセスは分割し1つのプロセス「窓口」としてまとめる必要がある。以下、これらについて述べていく。

6.1 流れの要求 (制御情報) や通知

貨物の目録は船が港につくにしたがって、税関に送られる。このシステムは倉庫の中の保管場所を海運会社に通知する。税関処理済みの貨物で引き渡し要求のあったものは倉庫からコンベアで出され代理店に渡される。

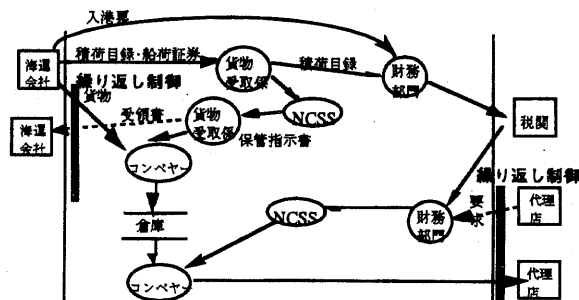


図9 データの流れの要求と通知

6. 2 内部オブジェクトの流れの記述

この例では内部オブジェクトは、倉庫の空き情報である。倉庫の空き情報は貨物情報と対になっている。この処理はNCSS内部のDFDとして記述される。

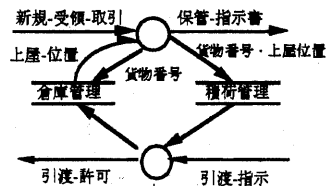


図10 内部オブジェクトのDFD

6. 3 プロセスの分割とデータフローの記録

組織に対応してプロセスの分割をする。また、分割に伴ってDFDを変更する。

いくつかのプロセスは処理の記録を履歴ファイルに残す。履歴ファイルに残っている記録を読むことで、貨物が今どのプロセスまで処理が進んでいるかがわかる(図11)。

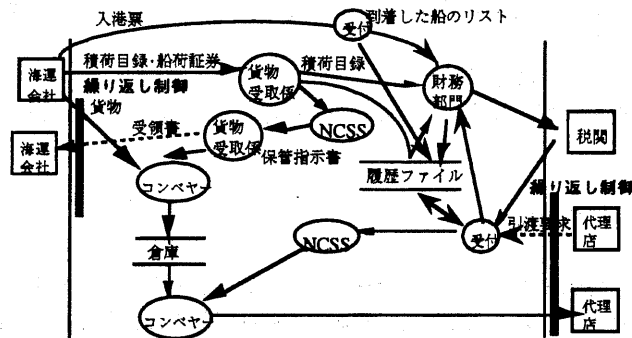


図11 組織に対応したプロセス分割と履歴情報の追加

7. むすび

本稿では、DFDの左右に事象トレースを記述する事象トレース付きDFDを提案した。事象トレース付きDFDは、事象トレースとDFDを同じ図で記述するため、DFDの処理手順が分かり易くなるとともに事象の因果関係がDFDで分かるよう記述できる利点がある。また、事象トレースの所に制御文の記述できるようにし、状態遷移図を記述することなく繰り返しを表現できるようにした。さらに、ビジネスプロセスのオブジェクトに着目した事象トレース付きDFDの段階的な設計手法を示した。本稿で提案した事象トレース付きDFDは事象とDFDとの関連を分かり易くするとともに、それを用いた段階的な設計手法は、システム分析者に設計の指針を与えるものである。

参考文献：

- (1) DeMarco, T. "Structured Analysis and System Specification", Yourdon Press, New York (1978).
- (2) Hawryszkiewicz, I.T. "Systems Analysis and Design", Prentice Hall, Inc. New Jersey (1994).
- (3) Rumbaugh, J. "Object-Oriented Modeling and Design", Prentice Hall, Inc. New Jersey (1991).
- (4) Jacobson, I. "Object-Oriented Software Engineering A Use Case Driven Approach", ACM press (1992).
- (5) Kendall, K. "Systems Analysis and Design", Prentice Hall, Inc. New Jersey (1995).