

## システムの準分解によるソフトウェア構造分析

熊 偉† 新 藤 久 和†

人工物の品質や生産性を規定する要因の一つに「構造」がある。形のあるものを設計する場合には、物理的な原理や法則が存在し、それらを考慮しつつ自然な構造が導かれる。一方、ソフトウェア設計においては、自然な構造を導出するための基礎となる原理や法則が十分明確にされていない。そのために、ソフトウェアの構造は設計者に左右されやすく、再利用などが進まない障害ともなっている。ソフトウェア工学の所産として、これまでさまざまな設計記述法が提案されており、最近ではケース・ツールとして活用されるようになってきている。しかし、ソフトウェアの構造を問題にするならば、記述するだけでなく構造を検討する基準が必要である。新藤はシステムに含まれる二つの異なる側面（要素）を取り上げてシステムを記述することを提案している。さらに、サイモンのシステムの準分解の概念を導入し、記述されたシステムを下位システムに合理的に分解することができることを報告している。本報告では、新藤の考え方をソフトウェアに適用することにより、ソフトウェアの構造を分析する方法を提案する。

### An Analysis of Software Structure Based on System's Near Decomposability

WEI XIONG† and HISAKAZU SHINDO †

Structure of a system can be one of major factors which may affect the quality and productivity of artificial products. In designing hardware products, physical laws and principles exist which lead designers to a reasonable structure. On the other hand, in software design such laws and principles have not yet been established. This means that the structure of a software system is easily dependent on designers, and as a result, software reuse has not been effectively promoted. So far, software engineering has proposed various description methods for software design. Recently, some of them have been employed as CASE tools. However, criteria or standard for structuralization of the description is very essential. Shindo proposed a system description method which uses two different factors of the system. Furthermore, applying Simon's idea of system's near decomposability to the description, he showed that the description could be practically decomposed. In this paper, by applying this idea, we propose a method for analyzing software structure.

#### 1. はじめに

人工物の品質や生産性を規定する要因の一つに「構造」がある。「形のあるもの」を設計する場合には、物理的な法則や原理が存在し、それらに基づいて自然な構造が導かれる。したがって、これらの法則や原理は構造化の基準を与えるものとして捉えることができる。一方、「形のない」ソフトウェアを設計する場合でも、構造化設計<sup>1)</sup>に象徴されるように、構造に着目することの重要性はいうまでもない。しかし、この場合の構造化の基準はそれほど明確ではない。そのために、ソフトウェアの構造は設計者の主観に依存しやすく、

出来映えを大きく左右する原因ともなっている。

よい構造を備えたソフトウェアを設計するためには、実現しようとしているソフトウェアをどのように記述し、それをいかに構造化すべきかが問題となる。前者の問題に対しては、ソフトウェア工学でも、これまでいろいろな設計記述法<sup>2),3)</sup>が提案され用いられてきている。たとえば、データ・フロー・ダイアグラム(DFD)やエンティティ・リレーション・ダイアグラム(ERD)などのように、いわゆるケース・ツールとして活用されているものもある。また、後者の問題に対しては、モジュール強度や凝集度といった指標が提案されている。こうした指標は、対象とするソフトウェアの詳細部分が明らかにされている場合には利用することができるが、設計の初期段階では当該ソフトウェアに対する知識や要求が不完全なため、必ずしも有効と

† 山梨大学工学部電子情報工学科  
Department of Electrical Engineering and Computer Science, Faculty of Engineering, Yamanashi University

はいえない。

新藤<sup>4)</sup>は、一般に人工物をシステムとしてとらえ、機械設計などで通常用いられている製図法からの類推により、システムを異なる側面とそれらの間の関連を二元表の組み合わせとして記述し、構造化することの意義について考察している。さらに、具体例として食堂を取り上げ、それに対する重要な側面として要求品質と品質要素に着目してそれらの関係を二元表で記述し、サイモン<sup>5)</sup>のシステムの準分解の考えを意識しながら、構造化を行い順次サブシステムに分解する方法について報告している。

本論文では、新藤<sup>4)</sup>のシステム記述と構造化の考えをソフトウェアに適用し、ソフトウェアの構造を分析/設計する方法を提案する。具体的には、ソフトウェアの重要な要素である、「機能」と「情報(データ)」に着目し、これらを二元表として記述した上で構造化する方法である。

従来、ソフトウェアの構造分析/設計においては、基本的には機能やデータを別々に取り扱って構造化を行っている。データ構造から制御構造を導こうとする方法などはその一例である。本報告では、機能も情報(データ)もソフトウェアを構成する基本要素としては同等と考える。さらに、それらの関連に着目することにより、両者を同時並行的に構造化するところに特徴がある。

例として、小規模な図書室の管理業務のためのソフトウェアを取り上げる。まず、このソフトウェアに要求される機能と、それぞれの機能に關係する情報(データ)を機能・情報関連表と呼ぶ二元表で記述する。次に、この二元表に構造化の手段として数量化<sup>3)</sup>類<sup>6)</sup>を適用する。その結果得られる各機能や情報に対するスコア(数量)を用いて、ソフトウェアの構造を分析/設計していく。

## 2. ソフトウェアの記述

「形のあるもの」の設計では「対象の構造・形状・寸法を一定のきまりに従って描いた図面」を設計図と呼び、「一定のきまり」が製図法として標準化されている。新藤<sup>4)</sup>は、製図法の本質は「複雑な立体を直交する三方向からみた側面の(形の)情報を展開し、相互の關係がわかるように図として表すところにある」と述べている。さらに、この考えを一般化して対象を無限次元の超立方体として図1(a)のようにモデル化し、重要な側面に着目して図1(b)のような二元表の連鎖として対象(システム)を記述することを提案している。

この考えに従えば、具体的な対象について、どのよ

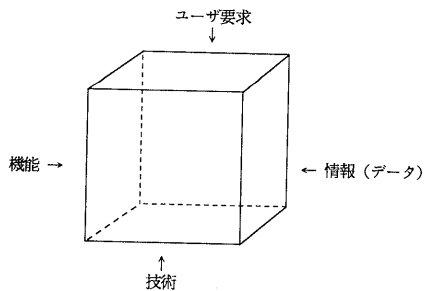


図1(a) システム超立方体モデル  
Fig. 1(a) Hypercube model of a system.

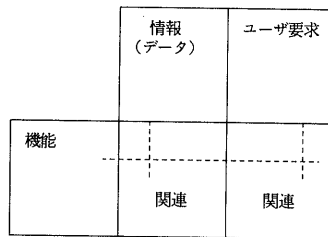


図1(b) 超立方体の展開図  
Fig. 1(b) Development of the system hypercube model.

うな側面を取り上げて記述するかが重要である。ここでは、ソフトウェアが対象であるから、「機能」と「情報(データ)」の二つの側面を取り上げて、機能・情報関連表として記述することにする。理由は、ソフトウェア工学における設計技法<sup>2),3)</sup>においても、これらは重要な要素として取り上げられてきているからである。

## 3. ソフトウェア記述の構造化

### 3.1 構造化の意義

機能・情報関連表として記述されたソフトウェアは、それだけで理解できる構造を有しているわけではない。一般には、気づかないまま見落とされている機能や情報があるであろうし、両者の関係も必ずしも正確でない場合もあるであろう。こうした不完全な記述をもとにして、十分理解できる形式に変換することが構造化の目的である。つまり、構造化の意義は、不完全な記述を完全にするとともに、複雑性を減少させて理解できるようにすることである。

例えば、図2(a)のような記述を考える。この場合、A~Fおよびa~hがそれぞれソフトウェアに要求される機能とソフトウェアに含まれる情報(データ)を表し、表の中の○印は機能と情報(データ)が関連していることを示すものとする。このとき、行同士あるいは列同士の交換は形式を変えるのみで内容は不変で

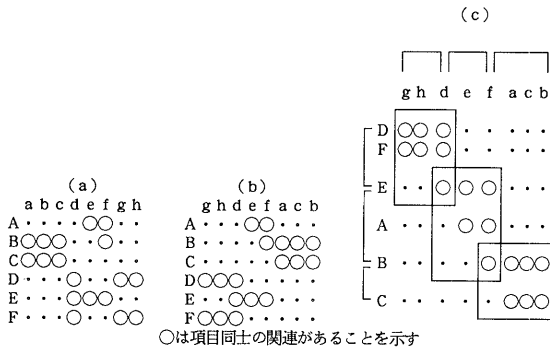


図2 二元表の構造化  
Fig. 2 Structuralization of 2-dimensional table.

ある。実際、列についての互換 (a, f), (b, h), (c, g), (d, g), (e, g), (f, g) を施すと、図2 (b) となり、さらに行についての互換 (A, D), (B, E), (C, E), (C, F) を施すと、最終的に図2 (c) のように変換することができる。これは、図2 (a) のように記述されたソフトウェアが図2 (c) のように網掛けを施した三つのサブシステムから構成されていることを示している。また、網掛けが重なっている部分はサブシステム同士のインタフェースに対応していると考えることができる。すなわち、機能については (DFE), (EAB) と (BC), 情報 (データ) については (ghd), (def) と (facb) がサブシステムを構成していると考えることができる。また、機能 E, B と情報 (データ) d, f はサブシステム間のインタフェースに関係していると考えられる。このように、機能と情報 (データ) を同等に取り扱うことにより、図2 (c) にみられるように、両者が同時並行的に構造化されていることに注目されたい。

この変換の過程が構造化であり、現実の問題ではこの過程で見落としている機能や情報を追加したり、サブシステムのレベル (階層) を調整することにより次第に完全な構造化されたソフトウェア記述を得ることができる。

ここで示した構造化の例は、サイモン<sup>9)</sup>が主張しているシステムの準分解と同等であり、サブシステム間にインタフェースが存在する一つの例になっている。

3.2 システムの準分解

サイモン<sup>9)</sup>は、複雑性を考察して、

- (1) 複雑性はしばしば階層的な構造をとる。
- (2) そして、階層的なシステムは、それぞれのシステムの個別的内容から独立した共通の性質をもつ。
- (3) さらに、階層こそ複雑性の構築に使用される構

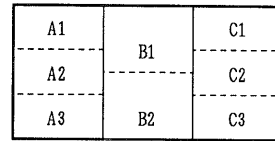


図3 建物の間取り  
Fig. 3 Layout of a floor.

表1 熱交換の係数マトリックス  
Table 1 Heat diffusion coefficient matrix for the layout.

	A 1	A 2	A 3	B 1	B 2	C 1	C 2	C 3
A 1	-	100	-	2	-	-	-	-
A 2	100	-	100	1	1	-	-	-
A 3	-	100	-	-	2	-	-	-
B 1	2	1	-	-	100	2	1	-
B 2	-	1	2	100	-	-	1	2
C 1	-	-	-	2	-	-	100	-
C 2	-	-	-	1	1	100	-	100
C 3	-	-	-	-	2	-	100	-

造的枠組みの中心的なものの一つである。と述べている。彼は、こうした考察に基づいて、ある種の階層的システムは近似的には準分解システムとしてうまくとらえることができると主張している。

サイモン<sup>9)</sup>は、準分解の具体的な例として、次のような部屋の間の熱交換システムを示している。図3のように、外壁からの熱を完全に遮断する一つの建物を考える。この建物は三つの大きな部屋からなり、大部屋同士の間の壁による熱の遮断は完全ではないが良好である。各大部屋は仕切りによってさらにいくつかの小部屋に分けられている。小部屋同士の間の仕切りによる熱の遮断はあまり良くない。このような建物の熱交換システムの熱の流れは、表1のようなマトリックスとして得られる。大部屋を一つのグループと見なすと、主対角線上に並んだ大きな値を持つ要素からなる正方部分行列が得られ、その外側の非対角要素は、すべてゼロまたは小さい値となる。ある小さい値をとり、これを非対角要素の上限とすることができよう。

複雑なシステムが、準分解により階層的にとらえられるならばシステムの理解を容易にするばかりではなく、設計にも有効に利用することができるであろう。

しかし、サイモンの示した例は記述そのものが準分解された形になっており、準分解の具体的な方法については触れられていない。また、この例は同一次元の

内部的な構造化を扱っている。

本論文では、異なる二つの次元を用いて記述されたソフトウェア記述の構造化を目的としている。また、準分解の具体的な手段として、新藤<sup>7)~10)</sup>と同様に数量化3類<sup>9)</sup>を用いることにする。

### 3.3 数量化3類

これまで記述を構造化することの意義とその基本的な方法について述べてきた。しかし、一般にソフトウェアを図2(a)のような形式で記述しようとする、機能や情報(データ)の数が多いため、試行錯誤で図2(c)のように変換することはほとんど不可能である。そこで、こうした変換を自動的に行うために、数量化3類を利用する。

数量化3類<sup>9)</sup>は、二元表として与えられた表の内部構造を分析するための手法である。任意個の水準をもつ二つの因子XとYについて、両者の関係が二元表の度数分布として与えられているものとする。因子XとYのそれぞれの各水準に適当な数量 $x_i$ と $y_j$ を対応させ、与えられた二元表の相関係数が最大になるようにこれらの数量を決定する。そのためには、固有値問題を解くことになり、それぞれの数量(スコアとも呼ばれる)は、固有ベクトルとして求められる。スコアの値によって、機能と情報(データ)の各項目を大きい順(あるいは小さい順)に並べ換えることにより、前節で示した記述の構造化を行うことができる。

得られた固有値のうち、数量化3類では定義から第2固有値以降の固有値・固有ベクトルを利用して構造を分析する。ここでは、第2固有値に属するスコアで機能・情報関連表を構造化することを考えるが、必要に応じて、第2および第3固有値に属するスコアを用いて作成した散布図を利用することもできる。

実際、図2(a)の記述において、関連を示す「○」

表2 固有値と固有ベクトル  
Table 2 The eigenvalue and eigenvector.

固有値番号	1	2	3
固有値	1.000	0.910	0.668
情報スコア	a 1.000	1.181	0.728
	b 1.000	1.181	0.728
	c 1.000	1.181	0.728
	d 1.000	-0.940	0.004
	e 1.000	-0.042	-1.988
	f 1.000	0.327	-1.203
	g 1.000	-1.290	0.801
	h 1.000	-1.290	0.801
機能スコア	A 1.000	0.149	-1.952
	B 1.000	1.014	0.300
	C 1.000	1.238	0.891
	D 1.000	-1.230	0.655
	E 1.000	-0.229	-1.300
	F 1.000	-1.230	0.655

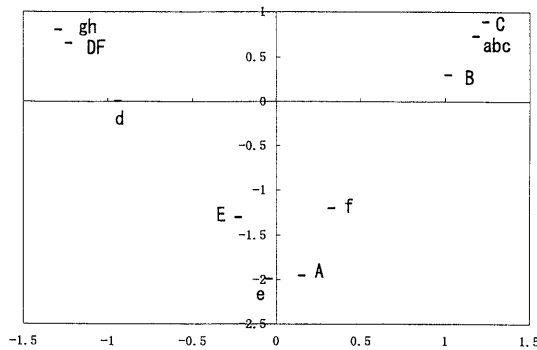


図4 図2の散布図  
Fig. 4 The scatter diagram of the Fig. 2.

印を数値「1」、関連がない箇所を「0」として数量化3類に入力した結果は表2のようになる。したがって、第2固有値に属するスコアを用いて、行項目(機能)と列項目(情報)を昇順に並べ換えることにより、図2(c)が得られる。さらに、第2および第3固有値に属するスコアから散布図を作成すると図4のようになる。これからも、全体が三つの部分(サブシステム)から成り立っていると考えることができる。この散布図を用いて通常のDFDを作成すると分かりやすい。

### 3.4 機能・情報関連表の構造化

本方法では、機能・情報関連表に数量化3類<sup>9)</sup>を適用して得られる新しい表の構造を検討し、再び数量化3類を適用するというサイクルを繰り返すことにより、徐々にソフトウェアの構造を導き出す。このようにして構造化された機能・情報関連表は、主対角線に上関係が集まるように項目が並べ換えられ、近くにある項目同士ほど関連性が強くなる。しかも、機能・情報関連表において、完全な対角行列にならず、それぞれのブロックが重なり合うところを「インタフェース」と考えることができる。ここでは、これらの性質を利用してソフトウェアをサブシステムに準分解しながら構造を分析する。

まず、機能・情報関連表の関連情報が集まった塊をなるべくほかの部分との独立性を高くするように切り出したブロックをサブシステムとしてとらえる。これは、システムの準分解の例で出てきた熱交換システムの大部屋に相当する。

しかし、一般的には、完全に独立なサブシステムというのは、希であり他と何らかのつながり、つまり、インタフェースが存在するはずである。これは先に述べたシステムの準分解の例での下位システム間の小さいが無視できない要素としてとらえることができる。

そして、分解の基準には、数量化3類により得られ

る機能と情報のスコアが活用できる。つまり、第2固有値に属するスコアの間隔の大きな部分に注目し、第2固有値と第3固有値に属するスコアを用いて作成された散布図を参考にしながら関連表を分解する。具体的には機能と情報（データ）それぞれの各項目のスコアを散布図にプロットすることにより、それらが塊をなしている部分をサブシステム（モジュール）として抽出することを考える。この方が表で検討するよりも情報量が多いだけ理解しやすくなる。それは表では一つの固有値に属するスコアとその順序関係しか用いていないからである。

さらに、このスコアの値に加えて、サブシステムの境界となりそうな辺りの個々の機能や情報の持つ意味についてよく考え、その上でどこで分解すればよいか検討する。

構造化の過程で、ユーザ要求の把握が十分でなければ、繰り返しの回数が増えるが、最終的に「対角化」を狙って構造化を進めて行くから、不十分な要求からでも利用することができると考えている。従来の方法では、要求分析が十分か否かを検討することは極めて難しい問題であるが、この方法では構造を視覚的に把握できるので、見落としなどを発見しやすい。また、従来の方法は同じ一つの次元に属する事象を構造化しているのに対して、本方法では、機能と（情報）データといった異なる二つの次元に属する項目とそれらの関連を同時に解析する。すなわち、この方法の特徴は次のようである。(1) 構造化の手順が明確に与えられる。(2) 機能と情報が同格であり、同時に両方が構造化される。(3) 設計者は分解されたサブシステム同士の関連の状態を視覚的にとらえることができる。(4) 構造を意識することにより機能や情報の漏れ、誤りなどが発見できる。なお、本方法を使用する段階としては、要求分析から構造設計ないし機能設計段階を対象としている。

## 4. 適用事例

### 4.1 対象ソフトウェア（システム）の概要

本方法を用いて分析を行ったソフトウェアは、筆者らが所属する学科の図書室の図書管理のためのソフトウェア（システム）である。

この図書室は、4万数千冊（単行本は約6千冊で、雑誌は約3万5千冊である）の蔵書があり、利用者数約400人の小規模な図書室である。蔵書の種類には単行本、雑誌および卒論、修論があり、その利用者は、本学科の教職員と学生である。

### 4.2 適用手順

ここでの手順は、一般的なステップとともにこの図書管理ソフトウェア（システム）についての具体的なステップを併記したものである。以下にその過程を順を追って述べる。

#### (1) 現状の調査

利用者と管理者に直接会い、現在の図書室の管理と利用状況を調べ、図書管理の仕組みや内容などを把握して、現状の分析を行った。

#### (2) ユーザ要求の収集

利用者および管理者から図書管理ソフトウェア（システム）に対してどんな要求があるのか聞き取り調査し、「どんな本があるかすぐに分かる」、「本が分類されて保管されている」、「返却日がすぐ分かる」など58項目のユーザ要求を収集した。

#### (3) 機能の抽出

ユーザから収集された要求は機能に対するものだけでなく種々の要求が混在している。そこで機能に対する要求を抽出するために、ユーザ要求の分類を行った。抽出された機能要求ごとにそれを実現するために、さらにどのような機能が必要であるかを考察し下位機能を抽出する。このとき、価値工学(Value engineering)の機能定義にならい、極力「名詞」+「動詞」という表現にする。

#### (4) データの洗い出し

機能ごとにどんな入力データが必要か、また、出力データが何かを検討して、データを洗い出した。

#### (5) 機能・情報関連表の作成

抽出された機能とそれに対応する情報（データ）を整理し、両者の関連付けを行って機能・情報関連表を作成した。一般にデータは入力データと出力データに分けられるが、ここでは入出力の区別はしていない。理由は、ある機能の出力データが他の機能の入力データとなり得るからである。

#### (6) 機能・情報関連表の構造化

機能・情報関連表に数量化3類を繰り返し適用し、構造を意識して各項目の分割、併合および項目間の関係を検討しながら関連表を分解していく。

①最初の機能・情報関連表ではいくつかのデータ項目（この例では、01図書名や02図書番号）がほとんどの機能項目に関連していることがある。このような項目はそのソフトウェア（システム）にとってキーとなるような重要なものであり、下位サブシステム間のインタフェースとしても考えられる。しかし、このような項目は、2元表に現れる構造を理解しにくくするため、これらを一時的に関連表から取り除くことにする。

表3 構造化された機能・情報関連表  
Table 3 Decomposed matrix of function-information relationship table.

機能	情報	2511231029093313014123213150506331608341726071903042228272120																			
		借返貸日 り出催 せ促 る状 量	返付用 者日 者者 番 号	返却用 約 登 止 利 滞 延 種 類 可 結 果	利 用 者 判 定 結 果	既 入 延 種 類 可 結 果	目 録 出 考 索 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	貸 出 可 結 果	
36借り出せる量を表示する		○																			
12貸出停止通知を印刷する		○																			
20貸し出されている利用者調べ			○																		
34貸出情報を出力する		○	○																		
09返却日を計算する		○																			
19返却日を延長する		○																			
10返却権限を印刷する		○																			
11利用者名・番号を入力する			○																		
18返却権限の概要を表示する			○																		
13返却日などが表示する			○																		
14返却日などのメモを印刷する			○																		
16予約を調べる			○																		
44利用者の訂正をする			○																		
46登録利用者を削除する			○																		
45登録利用者の情報を出力する			○																		
42新規利用者の情報を入力する			○																		
43既登録利用者の判定をする			○																		
15返却を確認する			○																		
17メールを送信する			○																		
04目次を表示する				○																	
47貸出可否を調べる				○																	
05図書種別を表示する				○																	
02図書を検索する				○																	
24参考文献を表示する				○																	
06分野別種別を出力する				○																	
26キーワードで図書を検索する				○																	
25キーワードを入力する				○																	
48貸出可能量を表示する				○																	
23著者名で図書を検索する				○																	
46貸出回数計算				○																	
37図書の貸出回数を記録する				○																	
29新規図書の紹介を出力する				○																	
27新規図書の情報を入力する				○																	
03登録図書の情報を出力する				○																	
21登録内容を変更する				○																	
22著者名を入力する				○																	
33注文した図書の情報を出力する				○																	
40備品の金額を入力する				○																	
32注文予定を出力する				○																	
31注文情報を出力する				○																	
41取支状況を計算し、表示する				○																	
38年間予算を入力する				○																	
39年度内購入図書金額を入力する				○																	
30図書の値段を表示する				○																	

また、機能・情報関連表において、関連データ数が多い機能項目について見直しを行い、下位機能への分解を行って、同一階層に含まれる機能の（概念）レベルを合わせた。同時に構造を意識して機能とデータの関連の追加、削除などを検討した。

②このような見直しを行った関連表に数量化3類を適用し、図5の散布図を参考にしながら関連表を表3のような三つの部分に分解した。この三つの部分の意味を考えると、それぞれ利用者管理、図書管理、経理のように解釈できる。システムの準分解の観点からは、これらは図書管理システムの下位システムと見なすことができる。

③各下位システムについて構造化を行っていく。ま

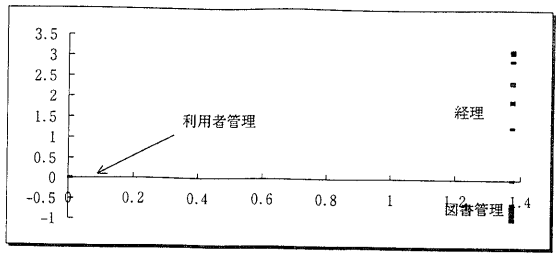


図5 表3の散布図  
Fig. 5 The scatter diagram of the Table 3.

ず、「利用者管理」について検討する。この部分を取り出して数量化3類を適用し、その散布図(図6)を参考にしながら表4のように分解した。各部分の意味を考

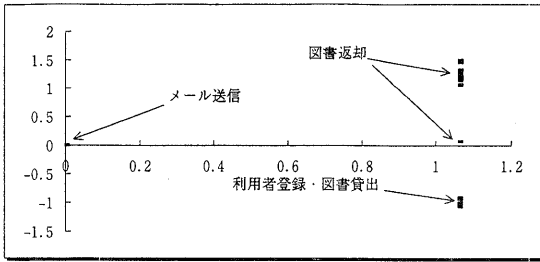


図6 表4の散布図

Fig. 6 The scatter diagram of the Table 4.

表4 利用者管理ブロックについての構造化

Table 4 Decomposed matrix of user management block.

機能	情報	141332233025311510091112 予メ既貸利借利返日返返貸 約！發出り用却却却出 状ル録借者出者延日催停 況文利解解所せ番滞 用者属号図書 者判定結果 の概要
	機能	15返却を確認する 16予約を調べる 17メールを送信する 43既登録利用者の判定をする 34貸出情報を出力する 36借り出せる量を表示する 46登録利用者を削除する 44利用者の訂正をする 45登録利用者の情報を出力する 20貸し出されている利用者調べる 42新規利用者の情報を入力する 11利用者名・番号を入力する 18返却延滞図書概要を表示する 19返却日を延長する 09返却日を計算する 14返却日などのメモを印刷する 13返却日などが表示する 10返却催促状を印刷する 12貸出停止通知を印刷する

えると、それぞれ「メール送信」、「図書返却」、「利用者登録・図書貸出」のように解釈できる。しかし、「メール送信」に比べて、「図書返却」と「利用者登録・図書貸出」のブロックは包含される機能の数が多いため、さらに構造化できると考えられる。そこで、「利用者登録・図書貸出」を取り出して数量化3類を適用すると、表5が得られる。表5において、データ項目「利用者番号」はほとんどの機能項目に関連しているので、先に述べたように一時的に取り除く。また、「貸出情報を出力する」項目は、ほかの機能項目に比べて、レベルが高いので、さらに下位レベルに分解することを考える。そこで、「貸出情報を出力する」という項目を「貸出者を表示する」、「貸出日を表示する」、「貸出量を表示する」の三つに分解する。このように分解すれば、

表5 数量化3類を適用した表4の真中のブロック

Table 5 Rearranged table of the middle block of Table 4 using quantification of type 3.

機能	情報	3231302523 既利貸借貸 登用り出 録者出情 利者所せ報 用属る 者量 判定結果
機能	機能	43既登録利用者の判定をする 45登録利用者の情報を出力する 20貸し出されている利用者調べる 46登録利用者を削除する 44利用者の訂正をする 42新規利用者の情報を入力する 11利用者名・番号を入力する 36借り出せる量を表示する 34貸出情報を出力する

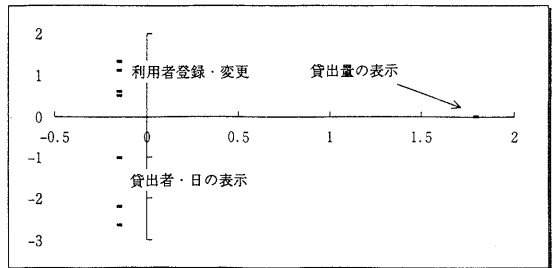


図7 表6の散布図

Fig. 7 The scatter diagram of the Table 6.

表6 構造化された表4の真中のブロック

Table 6 Decomposed table of the middle block of Table 4.

機能	情報	3230352536 既利貸借貸 登用り出 録者日量 利所せ 用属る 者量 判定結果
機能	機能	43既登録利用者の判定をする 45登録利用者の情報を出力する 46登録利用者を削除する 44利用者の訂正をする 42新規利用者の情報を入力する 50貸出日を表示する 49貸出者を表示する 36借り出せる量を表示する 51貸出量を表示する

「貸出者を表示する」は「貸し出されている利用者」を調べる項目と重複するので、後者を除く。

これらの検討を加えて、再度数量化3類を適用し、その散布図(図7)を参考にしながら「利用者登録・図

表7 準分解された機能・情報関連表  
Table 7 Decomposed function・information relationship table.

1次レベル 2次レベル 3次レベル 4次レベル			図書館システム																
			図書情報				利用者情報				経理情報								
情報 機能			貸出可能量	図書登録情報	図書検索情報	貸出回数	メール情報	利用者登録情報	貸出量	図書返却情報	収支状況	図書注文予定	0102 図書番号						
			34 貸出可能量	190403 新刊著者・新刊社名・新刊書名	1708 キーワード検索結果	33051606 貸出不可	26 貸出回数	07 図書分野	1413 予約状況	3230 既利用登録利用者判定結果	35 貸出日	253631 借り出せる量	15 返却延滞図書概要	1211 返却停止通知	100929 返却日付	2728 年間予算	222120 注文日	0102 図書番号	
図書館管理システム	図書管理	貸出可能量の算出	48 貸出可能量を表示する	○															
		図書登録	23 新刊図書の紹介を出力する	○															
			03 登録図書の情報を出力する 21 登録内容を変更する 27 新刊図書の情報を入力する 22 著者名を入力する	○○○○															
	検索	図書検索	25 キーワードを入力する		○														
			26 キーワードで図書を検索する 06 分野新編を出力する 02 図書を検索する 23 著者名で図書を検索する	○○○○															
		関連情報	24 参考文献を表示する 04 目次を表示する 47 貸出不可を調べる 05 図書種類を表示する	○○○○															
		貸出回数の算出	37 図書の貸出回数を記録する 46 貸出回数を計算する			○○													
	利用者管理	メール送信	15 返却を確認する				○												
			16 予約を調べる				○												
			17 メールを送信する					○											
		利用者登録・変更	43 登録利用者判定をする					○											
			45 登録利用者の情報を出力する 46 登録利用者削除する 44 利用者の修正をする 42 新規利用者の情報を入力する	○○○○															
図書貸出		貸出利用者の日の表示	50 貸出日を表示する 49 貸出利用者を表示する						○										
	貸出量の表示	36 借り出せる量を表示する 51 貸出量を表示する 11 利用者名・番号を入力する							○○○○										
図書返却	延滞図書概要表示	18 返却延滞図書の概要を表示する								○									
	催促・停止状の印刷	12 貸出停止通知を印刷する 10 返却催促状を印刷する									○○								
	図書返却日の出力	14 返却日などのメモを印刷する 19 返却日を延長する 09 返却日を計算する 13 返却日などを表示する										○○○○							
経理	収支管理	39 年間購入図書の金額を入力する 41 収支状況を計算し表示する 38 年間予算を入力する 40 備品の金額を入力する											○○○○						
	図書注文	31 注文情報を出力する 32 注文予定を出力する 33 注文した図書の情報を出力する 30 図書の値段を表示する 01 図書名・番号を入力する												○○○○					



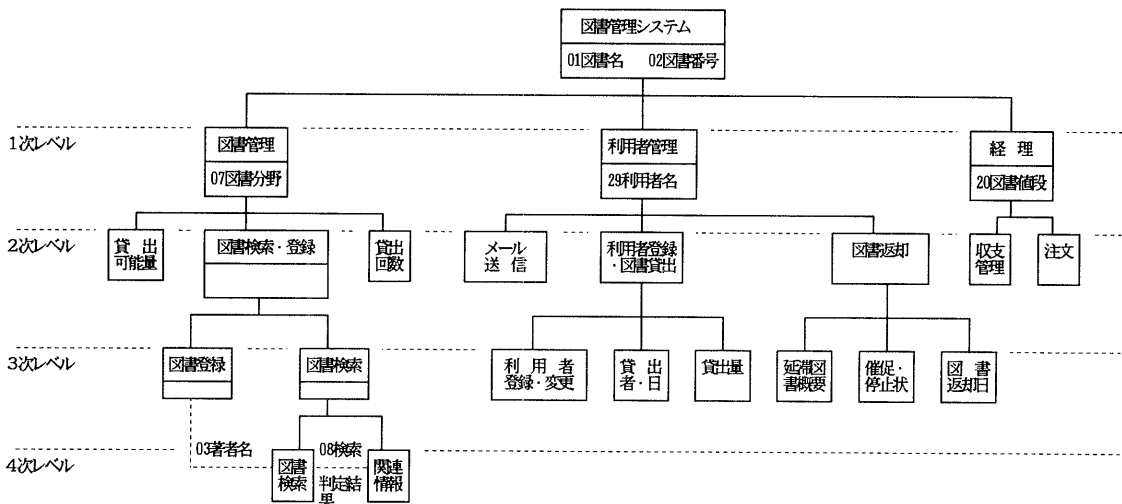


図8 ソフトウェア(システム)の構造概念図  
Fig. 8 A resultant structure of the software.

書貸出」が三つのブロックに分かれた表6が得られる。

④同様な検討を他のブロックについても行うと、関連表は最終的に表7のように分解される。分解された各ブロックは図書館管理ソフトウェア(システム)のサブシステムとして考えられる。各ブロックの共通データ(例えば、表7の中の“01 図書名”, “02 図書番号”)は各サブシステムのインタフェースとしてとらえる。または各ブロックのつながりもインタフェースとして考えられる。(例えば表7の中の“08 検索判定結果”データは“図書館検索”サブシステムと“関連情報”サブシステム間のインタフェースとして考えられる)。また、表7の左欄から見ると機能が階層化されており、表頭の上部からみるとデータも階層化されている。

#### (7) 構造仕様の作成

表7から、図8のようなサブシステムの構造概念図を作成した。さらに、サブシステムの構造と制御、内部・外部インタフェースを規定する構造仕様を作成する。

#### (8) DFDとの併用

データ・フロー・ダイアグラム(DFD)を作成する場合に、レベリングは準分解された階層構造に基づき、各レベルのDFDの作成は対応している散歩図を用いる。このようにして構造が理解しやすい配置になるDFDができる。図9は「利用者登録・図書貸出」サブシステムについてのデータ・フロー・ダイアグラム(DFD)であり、下位サブシステム間のデータの流れを示している。また、このDFDを用いて構造の妥当性が検討できる。たとえば、ステップ6の3で「利用者登

録・図書貸出」を分解するときには、「利用者登録・変更」、「貸出者・日」、「貸出量」の三つの下位サブシステムに分解された。しかし、DFDの場合には、「貸出者・日」と「貸出量」を合わせて「図書貸出」で表示し、「利用者登録・変更」のレベルに対応すると考える。「貸出者・日」と「貸出量」は「利用者登録・変更」のレベルに比べて細かすぎると考えられる。さらに、DFDとの併用によりデータ項目の追加を行うことができる。たとえば、「利用者の訂正をする」について「訂正項目」というデータが必要であることが分かった。

#### (9) 詳細仕様の作成

構造仕様とデータ・フロー・ダイアグラム(DFD)を用いて、各サブシステムの機能、入出力、内部処理手順、内部データを定義する詳細仕様を作成する。

### 4.3 考察

ここでは、この適用事例より得られたノウハウなどを述べる。

(1)機能レベルについて、ユーザー要求から機能を抽出する時点で、機能レベルを統一することが難しいが、構造化を行いながら機能レベルを整えることができる。

(2)機能・情報関連表において、非常に関連項目数が多いデータ項目を関連表から一時的に取り除き、構造化を行っていくとわかりやすい。

(3)構造化では、項目の見直しが機能面からばかりでなく、データ面からも項目の見直しを行っていくことが大切である。

(4)サブシステムへの分解において、数量化3類で

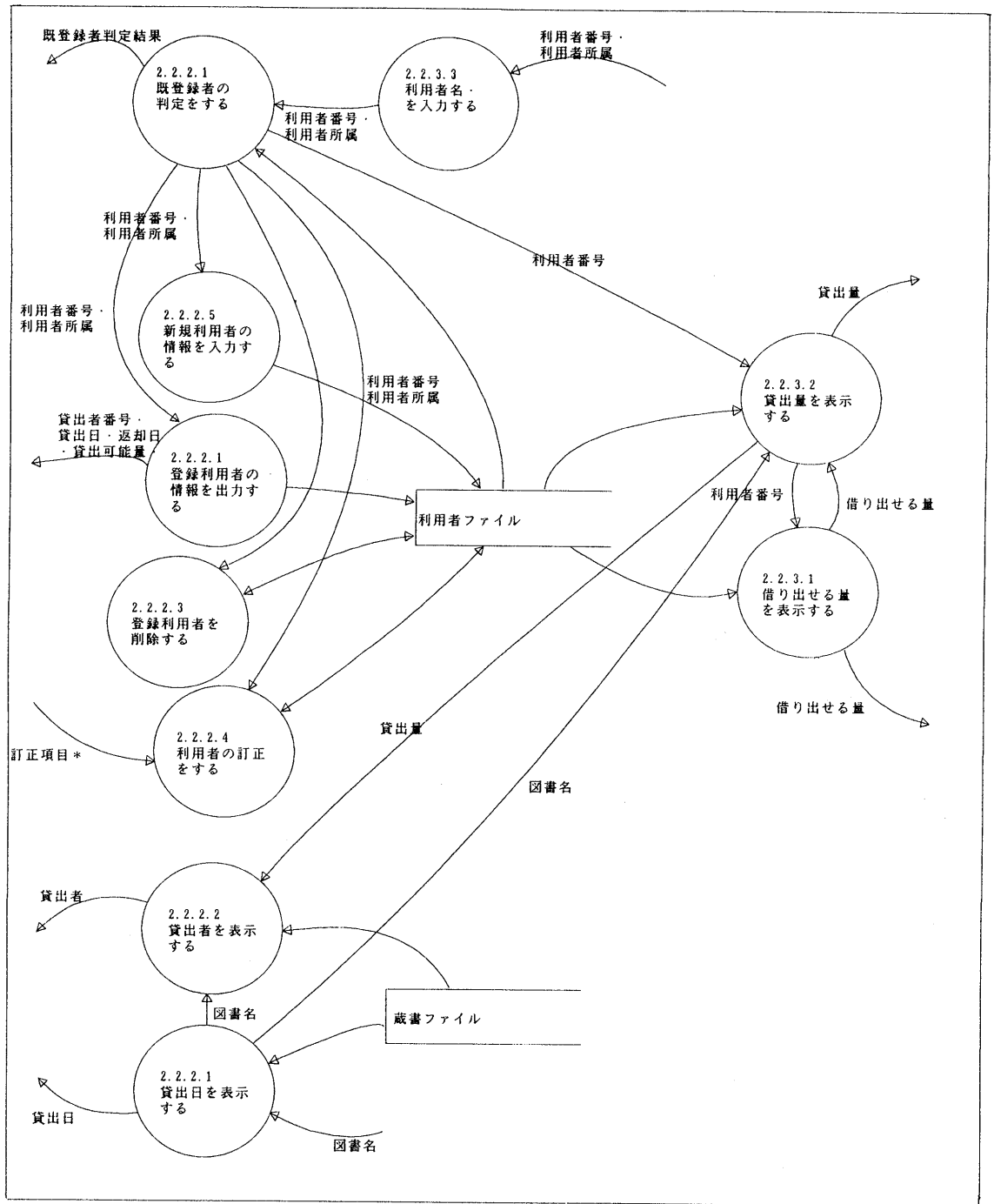


図9 「利用者登録・図書貸出」サブシステムのDFD  
 Fig. 9 DFD of user registration and borrowing subsystem.

得られるスコアの値を用いるが、スコアの値が飛び離れた項目間は分解できることが多いので目安にすることができる。また、このスコアの値からだけでなく、項目の意味を合わせて考える必要もある。

(5)データ・フロー・ダイアグラム (DFD) を併用することで、各サブシステム間の情報の流れを示し、構造を検討し修正することができる。さらに、データ項目の追加を行う。

適用事例から、以上のような構造化のためのノウハウを得ることができた。今後は、適用事例を増やし、さらにノウハウを蓄積することが必要である。

## 5. おわりに

本論文では、新藤<sup>4)</sup>のシステム記述と構造化の考え方をソフトウェアに適用することにより、まず、機能・情報関連表と呼ぶ二元表を用いたソフトウェアの記述法について述べた。次いで、機能・情報関連表に数量化3類を適用し、システムの準分解に基づいて構造化を行うソフトウェア構造分析法を提案した。さらに、この方法を実際のソフトウェアに適用し構造化を行った。

今後は、本方法をソフトウェア分析/設計法として活用して行くために、実際にソフトウェアを開発して評価することを考えている。また、実時間システムの場合についての取り扱いも今後の課題としたい。

**謝辞** 本研究を進めるに当たり山梨大学栗原光信教授、山下茂技官には多大な援助を賜った。また、山梨大学学生一ノ瀬光裕君にはデータ整理やDFDの作成をしていただいた、記して謝意を表す次第である。

本研究の一部は日本電気(株)の奨学寄付金によるものであり、ご支援に感謝したい。

## 参考文献

- 1) 花田收悦(編): ソフトウェアの仕様化と設計, 日科技連出版社, 東京 (1986).
- 2) 菅野文友(編): ソフトウェアの生産技法, 日科技連出版社, 東京 (1986).
- 3) J. マーチン, C. マックルーア(著), 国友義久, 渡辺純一(訳): ソフトウェア構造化技法—ダイアグラム法による, 近代科学社, 東京 (1986).
- 4) 新藤久和: システム記述手法としての品質表概念の一般化とその準分解による構造化, 品質, Vol. 23, No. 3, pp. 95-104 (1993).
- 5) Simon, H. A.(著), 稲葉元吉, 吉原英樹(訳): 新版システムの科学, パーソナル・メディア社, 東京 (1988).
- 6) 小林竜一: 数量化理論入門, 日科技連出版社, 東京 (1981).
- 7) 新藤久和: 品質表と構造化, 第2回品質機能展開

シンポジウム発表報文集, pp. 85-92 (1992).

- 8) 新藤久和, 吉沢 正, 宮島正明: サービス業における品質表と数量化第3類の適用, 品質, Vol. 13, No. 3, pp. 53-60 (1983).
- 9) 小高守道, 熊 偉, 新藤久和: 機能・情報関連表に基づくソフトウェア構造分析, 第13回ソフトウェア生産における品質管理シンポジウム発表報文集, pp. 171-178 (1993).
- 10) 熊 偉, 一ノ瀬光裕, 新藤久和: ソフトウェアシステムの記述法とシステムの準分解に基づく構造化, 日本機械学会第71回通常総会講演会論文誌, pp. 315-317 (1994).

(平成6年1月18日受付)

(平成7年1月12日採録)



熊 偉 (正会員)

1963年生, 1985年中国華東船舶工業学院機械工学科卒業。1987年同大学院双学士コース管理工学専攻修了。同年同大学管理工学科助手。1992年山梨大学大学院博士後期課程入学, 現在に至る。ソフトウェア設計論, ソフトウェア品質管理などの研究に従事。品質管理学会会員。



新藤 久和 (正会員)

1970年山梨大学工学部電子工学科卒業。1972年山梨大学大学院電子工学専攻修士課程修了。同年山梨大学助手工学部計算機科学科。1986年同講師を経て, 1993年同助教授, 電子情報工学科, 現在に至る。ソフトウェアの品質管理, データ解析, 品質機能展開等の研究に従事。工学博士。1977, 78年米国カンザス州立大学にて, 日本学術振興会日米科学協力事業による共同研究。1981年から83年, 情報処理振興事業協会プロジェクトリーダー。1985年から87年, 同コンサルティング委員。1982年SQC賞(品質管理誌編集委員会)。1983年論文奨励賞(日本品質管理学会)。日本品質管理学会理事, 編集委員会委員長。粉体工学会, スポーツ方法学会等会員。デミング賞委員会委員。