

知識型統合ソフトウェア VCAP†

—システム設計および異種統計パッケージ制御列の自動生成—

松田孝子^{††} 鈴木篤^{†††} 田中信行^{††††}

VCAP は異種ソフトウェアをユーザ視野から統合するシステムである。事例として開発母体の異なる統計パッケージ SPSS と STATPAC を取り上げ、オブジェクト指向の概念により設計開発した。統合対象ソフトウェアを従来の手続き中心から情報中心に把え直し、利用上の知識をシステムに内蔵してシステム駆動の核として用いることにより、異種プログラムを同一のユーザインタフェースで制御する。また従来人間がエキスパートとして果してきたエンドユーザ指導の役割を VCAP に行わせる。VCAP に内蔵する情報は、統合対象のアプリケーションプログラムのほかにデータベース管理システム COOD および論理型言語 ShapeUp などのプログラム群、解析対象のデータファイル・データベース、マニュアルに記載されている使用規則を格納したデータベースおよび利用用途で習得蓄積する使用法・使用例からなる知識ベースである。処理プロセッサを、統計解析の仕事を反映した対象プロセスに分割し、さらに処理片に細分した。これらの処理片をメッセージ授受で連結し対象プロセスを連係して仕事を遂行する。VCAP は格納知識を用いて対話者を誘導しながらその要求を受け取り処理手順を自動生成して、ユーザの一連の処理操作を行う。VCAP を介することによりユーザはソフトウェア個別の技術を習得する労力から解放され、ソフトウェアへの対応を技術中心から問題中心に改善できる。

1. ま え が き

機能分担型の計算機環境が実現する中で、ソフトウェアの量的増加と機能の多様化が著しい。また、他のソフトウェアとの関係を考慮せず独立に開発されるソフトウェアの単独な流通も盛んである。一方、利用の大衆化が進み、コンピュータの専任要員ではないエンドユーザが増え、各自がコンピュータを操作して仕事を遂行するようになってきた。このような状況の中で、ソフトウェアを利用する場ではつぎのような問題が起こっている。

- 遂行する仕事の処理過程で最適なソフトウェアを選択したいが、その手掛かりとなる利用上の知識を手にするのが困難である。
- 目的に合わせてソフトウェアを適宜選択し利用したいが、そのために習得すべき規約が多すぎる。また、利用側からの体系化が不十分である。
- 処理に応じて選択したソフトウェアを連係して一つの仕事にまとめる場合に、ソフトウェア間のインタ

フェースが確立されていないことが多い。

このような問題を解決するには、ユーザ視野からの体系化に基づくソフトウェアの統合が必要である。従来、データの統合¹⁾に端を発してデータベース技術が開発され、分散リソースの共用を目指してコンピュータネットワーク²⁾が実現した。そして、これらの技術の上に分散データベース、分散プログラム、プログラムとデータベースの統合³⁾⁻⁷⁾が研究開発されてきた。また最近では、オフィスシステムにおける統合ソフトウェアが研究され実用化されている⁸⁾。しかし、これらはデータや通信メディアあるいはプログラム機能を統合するもので、前述の問題を解決しない。

我々は、この問題を解決するために、新たな統合方式として知識中心の統合方式を提案し、その実現例として、開発母体の異なる異種のアプリケーションプログラムを対象とする知識型統合ソフトウェア VCAP (Virtual Control of Application Program Packages) を設計開発した^{12),13)}。VCAP は、ソフトウェアを利用する上での知識を知識情報処理^{9),10)}を適用してシステムに組み入れ、それをユーザに提供するとともにシステム駆動の核としても用い、異種のソフトウェアの統合をはかるものである。システム設計にはオブジェクト指向の概念¹¹⁾を適用した。事例として異種統計パッケージを取り上げた。

本論文では、次章で VCAP の概念設計を述べ、3章で統計パッケージの統合を事例とするシステム設計

† Integrated Software VCAP with Knowledge Information —System Design and Automatic Construction of Control Sequences for Heterogeneous Statistical Packages by TAKAKO MATSUDA (Computer Center, Tohoku University), ATSUSHI SUZUKI (Research Center for Applied Information Science, Tohoku University) and NOBUYUKI TANAKA (Nippon Jimuki Co., Ltd.).

†† 東北大学大型計算機センター

††† 東北大学応用情報学研究センター

†††† 日本事務器(株)

について述べる。4章では内蔵知識を適用して統計パッケージの制御カードデッキを自動生成する方法について詳述する。5章では本方式について評価する。

2. 統合ソフトウェア VCAP の概念設計

2.1 ソフトウェア利用の現状分析

エンドユーザがコンピュータを用いる場合には、アプリケーションソフトウェア aS を準備する。 aS は、プログラム P 、データ D および制御指令 C からなる組で、次式のように表す。

$$aS = (P, D, C) \quad (1)$$

$$P = \{P_s, P_L\}$$

$$D = \{D_s, D_L\}$$

$$C = \{C_B, C_A\}$$

ここで、 P_s と D_s は利用者個人のプログラムとデータ、 P_L と D_L はライブラリー化されたプログラムとデータ、 C_B は基本的な制御指令、 C_A は応用依存の制御指令である。この区分要素の組み合わせによって、ソフトウェアの利用形態を表すことができる¹³⁾。ここでは、アプリケーションプログラムの利用を取り上げ

$$aS_{AP} = \{C_B, C_A, D_s/P_L\} \quad (2)$$

とする。式(2)の右辺の括弧内の斜線の左側は利用者が準備するもの、右側は運用システムの中に既登録のものを示す。

式(2)で表す形態の中から最も代表的な統計処理を取り上げ、その仕事を分析するとつぎの七つの処理に区分できる。

- ①データの整理
- ②解析手法の決定
- ③統計パッケージ・手続きの選択 (P_L の選択)
- ④制御カードの組み立て (C_B, C_A の作成)
- ⑤データの投入 (D_s の作成)
- ⑥統計計算
- ⑦結果の出力

このうち現在直接コンピュータを用いているのは⑤から⑦までの部分で、①から④までは机上作業で行っている。特に③と④は既存のプログラムに関する利用上の知識や規約を参照、適用することであり、そこでは経験者が定常的に指導要員としての役割を分担している。

一方、コンピュータの利用形態は、バッチ処理指向からオンライン処理指向に様相を変化して、ファイルを活用した利用法が普及し、旧来のカード利用が姿を消しつつある。データベース化などによりファイル上

のデータを仕事の中心に据え、目的に応じてプログラムを選択し、遠隔の端末からの指令で処理を連係させて一連の処理系列にまとめて実行しようとする。この場合、ソフトウェアを利用するための種々の情報が関連性をもって容易かつ即時に獲得可能であるかどうか極めて重要な問題となる。そのため、運用側にとっては情報の収集、整備、提供作業が大きな比重を占めつつある。

統計パッケージは機能上の特徴に加え固有の制御方式やファイル編成を用い、個々独立に閉じた世界を形成している。また、これらの利用に関連する情報は、マニュアルやユーザガイド等資料類の中、運用者の管理記録の中および利用者の経験の中に散在しており、量的にも極めて多い。このため、エンドユーザが各プログラムの特質を活かして複数のパッケージを横断的に使用する場合には、多くの分散した情報を各自収集し組み合わせて適用しなければならず、円滑な利用に困難をきたしているのが実情である。

2.2 VCAP の概念設計

異種のソフトウェアを統合する場合、それらの開発仕様を変更して新たなアーキテクチャの下に再構成し直すことは困難である。既存の仕様を変更することなく異種性問題を解決するには、共通仕様を設定して各ソフトウェアからの変換プログラムを個別に作成する方法があるが、この場合、統合対象のソフトウェアの数に比例して開発管理作業量が増え、提供機能は共通仕様へのマッピングの際に一般に縮小される。

そこで、VCAP では、2.1 節で述べた分析結果を踏まえて、つぎの方針に基づく設計を行うことにした。

(1) 従来プログラム中心にとらえてきたソフトウェア体系を利用に関連する情報中心にとらえ直して設計する。

(2) 利用者が各自の仕事を効率的に遂行できるようにする。また利用の容易性をはかる。

(3) 従来指導要員が分担してきた仕事のうちで機械化可能な部分を VCAP に組み込む。

(4) ソフトウェアを利用する上での経験的な知識を収集し体系化して VCAP に内蔵し、その消失を防ぐとともに円滑な活用をはかる。

(5) マニュアルやユーザガイドに記載されていた事項をできるだけ VCAP に内蔵し、ユーザが参照する印刷物の量を極力減らす。

(6) 利用に関連する情報を、ソフトウェアの開発側がトップダウンに構築するものと、利用側が使用経

験の中でボトムアップに集積するものに分けて体系化する。

(7) 異種のソフトウェアを同一のユーザインタフェースによって制御する。VCAP は内蔵情報を用いて、対話者の要求に応じた実際の処理を生成する。

(8) ユーザの仕事の流れに対応する処理対象を設定しておき、ユーザ要求に応じて VCAP が処理対象を選択し実行する。また、ユーザが要求する一連の処理系列を VCAP の中で自動的に連係できるようにする。

VCAP に内蔵する情報 Σ_1 を、式(1)の三つの構成要素に利用上の知識を加えたものに対応して、つぎのように区分した。

$$\Sigma_1 = \{P, D, C, K\} \quad (3)$$

ここで、 P はオペレーティングシステム上の既存の各種処理プログラム、 D はアプリケーションプログラムを適用して解析するファイル上のデータまたはデータベース化したデータとする。 C はプログラムの実行制御指令を作成するための規則からなる制御情報で、従来はプログラムごとのマニュアルやユーザガイドに記載されていたものとする。 K は利用に関するノウハウからなる知識情報で、従来はユーザ個々の経験として保有していたものとする。 C に属する情報も知識として扱ってもよいが、VCAP では、アプリケーションプログラムの開発者がトップダウンに作成可能である C を狭義の知識情報 K から除外して構築管理体系の確立されたデータベースとし、VCAP で分類する K は使用例等のエンドユーザが各自の使用経験の中からボトムアップに習得し集積する知識だけからなるものにした。このような K は運用側で収録公開するものに加えて、ユーザ各自の知識としてもつことが一般的に行われるので、これら二つの知識を接合して使用することも起こる。そこで、共用の知識と私用の知識の内蔵形式および操作を同一、柔軟、単純しておく必要があり、VCAP では論理型言語による述語表現を用いることにした。

コンピュータによる処理の基本（処理片）は、対象とするソフトウェア片に入力メッセージを与え、出力メッセージを得ることであると規定できる。このソフトウェア片はある処理操作を指示するプログラムとそこで使用するデータの組からなる。前者を作用片 p 、後者を情報片 d とし、処理片を

$$\text{in}[m] \rightarrow [p|d] \rightarrow \text{out}[\bar{m}] \quad (4)$$

のように表す。ここで、 m および \bar{m} はそれぞれ入力

および出力メッセージ、 $[p|d]$ は処理片における対象ソフトウェア片である。 p は P に属する要素、 d は K, C あるいは D に属する要素である。一般に、このような処理片の複数連鎖によってまとまった処理を達成し、処理の連係によって一つの仕事を完成する。そこで、VCAP の処理プロセッサを、遂行する処理に対応する対象プロセスに分割し、それらをさらに処理片に細分して実現した。すなわち、処理プロセッサを式(4)の形式の連鎖によって記述し、プログラムへの変換を行った。VCAP の処理は、内蔵情報を $K \rightarrow C \rightarrow \{P, D\}$ の順に選択、合成、適用して、ユーザが入力する必要最小限の指示から実際の制御系列を自動的に生成し、対象プロセスを連係実行して、ユーザの要求する最終結果を出力することである。

3. 異種統計パッケージの統合

3.1 システムの構成

VCAP の事例として ACOS-1000 上の異種統計パッケージ SPSS (SPSS Inc. 製)^{14), 15)} と STATPAC (日本電気製)¹⁶⁾ を取り上げ、VCAP-1 を開発した。データベース管理用に COOD¹⁷⁾ を、知識情報処理用に ShapeUp¹⁸⁾ を用いた。システム構成を図1に示す。式(3)の Σ_1 をつぎのものとした。

$$P = \{\text{SPSS, STATPAC, COOD, ShapeUp, others}\}$$

$$D = \{\text{O-file, V-file, A-file, W-file}\}$$

$$C = \{\text{CIJ, CIC}\} = \text{CI}$$

$$K = \{\text{KBA, KBU}\} = \text{KB}$$

O-file は統計解析を施す原データファイルで、1 ケース分のデータをいくつかのデータ変数からなるレコードとして構成する順編成ファイルである。A, W-file は SPSS と STATPAC にそれぞれ個別に依存する固有形式のファイルである。V-file はこれらの異なるファイルを連結するための仮想データファイルで COOD データベースとして実現した¹²⁾。SPSS のもつデータ変数のラベルは COOD のデータ定義におけるアイテムの説明として表現した。STATPAC はデータ変数を番号で指示し、SPSS のようなデータ定義機能をほとんどたないので、V-file を用いることによりデータ操作が行いやすくなった。また、統計データの中の missing 値は COOD の null 値で表現した。VCAP を介して取り扱うデータは、V-file の中に格納して統計パッケージに依存しない方法で共通に管理し、実際の処理時に目的のパッケージの A-file

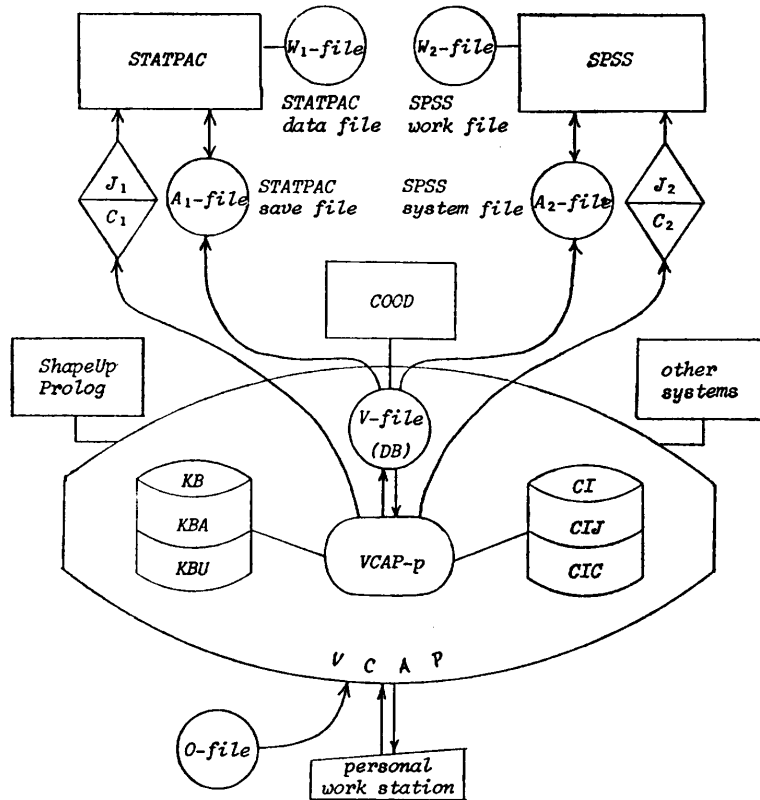


図1 VCAP-1のシステム構成
Fig. 1 System configuration of the VCAP-1.

に自動的に転送して統計解析の対象とすることにした。

SPSS と STATPAC はいずれもバッチ処理，カードベースの仕様をもつ統計パッケージで，SPSS はキーワード方式によるほぼ自由形式，STATPAC は完全な固定形式の手続き制御カードを使用している。また，それぞれのジョブ制御カードやファイル機番も独自に決められている。そこで，ジョブ制御カードと手続き制御カードの一般規則を，それぞれ CIJ, CIC として制御情報データベース CI の中に格納した。また，統計パッケージの照会情報と使用例情報を知識ベース KB の中に格納した。このうち，KBA にはパッケージ手続きを選択するための手掛かりに用いる情報を，KBU にはパッケージ手続きを実行するための使用例制御カード列の情報を蓄積している。CI と KB については 3.2 節でさらに述べる。

処理プロセッサ VCAP-p は，KB の情報を用いてユーザ（対話者）を誘導しながら

らその処理要求を受け取り，実際の処理手順を編成し，CI の情報を用いて適用するパッケージ用の制御カード列 J_i (ジョブ制御カードデッキ，式(2)の C_B) および C_i (パッケージ内手続き制御カードデッキ，式(2)の C_A) を自動生成し，統計パッケージの実行および結果の出力までのユーザの操作を自動的に遂行する役割を果たすものである。VCAP-p については 3.3 節でさらに述べる。

3.2 制御情報および知識情報の形式化

制御情報を収録する CI を図 2 のデータ構造からなる COOD データベースとした。手続き制御カード用の CIC は CICSPSS と CICSTAT という名称のパッケージ別のテーブルとし，ジョブ制御カード用のテーブル CIJ と共に同じ形式の表構造とした。CI 中の格納項目は，制御カード名，表題，機能，制御カードの一般形式，パラメータなどの記述説明である。このうち制御カードの一般形式を記述するために表 1 の記号を定めて用いた。例 1 に，SPSS の FREQUENCIES (度数分布) と CROSSTABS (クロス表分析)，STATPAC の COMP1 (数量化 I 類) と D-CREF (データ

```

DDL;
DATABASE CI : Control Information of VCAP;
TABLE CIJ : Job Control Language of Packages;
NO (I3) unique : Serial no. of card set;
CNAME (A8) : Card set name;
.....
TABLE CICSPSS : Control Card of SPSS;
NO (I3) unique : Serial no. of control card;
CNAME (A15) : Card name;
TITLE (A70) : Title of card;
FUNC(10) (A70) : Function of card;
FORM(10) (A70) : Card form of parameter list;
PARAM(20) (A5) : Parameter;
EXPL(20) (A70) : Explanation of parameter;
IFMT(20) (A5) : Input format of parameter;
REF(20) (A5) : Reference;
TABLE CICSTAT : Control card of STATPAC;
NO (I3) unique : Serial no. of control card;
CNAME (A6) : Card name;
.....
END-DDL;
    
```

図2 制御情報 CI のデータ構造
Fig. 2 Data structure of the control information CI.

表 1 制御カードの記述記号
Table 1 Symbols of the description rule of control card forms in VCAP.

分類	記号	意味
位置付け	!n	第 n 桁に位置付け (単一カードの場合)
	\n	第 n 桁に位置付け (複数カードの場合)
	~	空白を一つ挿入
パラメータ	◇a	パラメータ a
括弧および繰り返し	[p]	p 省略可
	{p1 p2}	p1 または p2
	[p1 p2]	p1 または p2 (省略可)
	{p}; n	p の n 回繰り返し
	[p]; n	p の n 回繰り返し (省略可)
	{p}; *	p の不定回繰り返し
[p]; *	p の不定回繰り返し (省略可)	

生成) の制御カードの一般形式の記述例を示す。

例 1)

```

FREQUENCIES \16 GENERAL=◇v1
[OPTIONS \16 ◇op]
[STATISTICS \16 ◇st]

CROSSTABS \16 TABLES={◇v1}~BY~
◇v2}; */; *
[OPTIONS \16 ◇op]
[STATISTICS \16 ◇st]

COMP 1 !7 ◇d ◇n[◇nl]◇v1[◇a1 ◇a2 ◇a3]
\7 {◇v}; n
[+ !7 ◇d1 ◇a4 ◇a5]
[* !7 ◇d2 ◇c]

D-CREF ◇d ◇mis ◇c
PARAMS ◇nv !16 ◇ns
NAME
[!7 ◇no ◇v]; *
VARIBL
(V)
    
```

知識情報を収録する KB を図 3 のように、パッケージごとの管理情報 A_S 、パッケージ手続きの内容照会のための情報 A_R およびパッケージ手続きの使用例題情報 A_E からなる KBA と、使用例の手続き制御カード列 I_c およびジョブ制御カード列 I_j からなる KBU に分類して、ShapeUp の述語として格納した。 KBU の中の使用例制御カード列をつぎの規則で記述した。

$$I: \{ \nu \{ [, \rho] \} \dots ; \} \dots ; \quad (5)$$

$KBA = \{ A_S, A_R, A_E \}$
 $A_S (n, v, d, w, t_p, t_j, h)$ パッケージの管理情報
 $A_R (k, n, p)$ パッケージ内手続きの内容照会情報
 $A_E (n, p, e, r_c)$ パッケージ手続きの使用例題情報

$KBU = \{ I_c, I_j \}$
 $I_c (r_c, r_j, c)$ 使用例の手続き制御カード列
 $I_j (r_j, j)$ 使用例のジョブ制御カード列情報

n : パッケージ名, v : パッケージのバージョン,
 d : サービス開始日, w : ユーザニュースの掲載番号,
 h : ソフトウェア開発者名
 t_p : 制御情報データベース CI 中の手続き制御カード規則を格納しているテーブル名
 t_j : 制御情報データベース CI 中のジョブ制御カード規則を格納しているテーブル名
 k : 検索用キーワード, p : パッケージ内の手続き名,
 e : 使用例題の説明
 r_c : 使用例手続き制御カード列の識別名
 r_j : 使用例ジョブ制御カード列の識別名
 c : 手続き制御カード列の例
 j : ジョブ制御カード列の例

図 3 知識情報 KB の分類と表現

Fig. 3 Classification and representation of the knowledge information KB.

I は使用例の識別名, ν は CI 中の項目 CNAME に格納している制御カード名, ρ はこの例で使用するパラメータ名とその実際にとる値を記述するもので

$$a[\alpha] [=r] \quad (6)$$

の形式とした。ここで, a は CI 中の項目 PARM に格納しているパラメータ名, α はこの使用例記述の中だけで有効な a の代替名を定義する。 r にはパラメータ a のとる実際の値を引用符で囲んだ文字列として書くか, またはこの使用例記述の中で定義している代替名 α を書く。 α を指定した場合はそれを定義しているパラメータ a と同じ値をとることを示す。

式(5)および(6)の超記号は, [] が括弧内省略可, { } … が括弧内繰り返し可を意味する。その他の記号はそのまま記述する。 r が省略されている場合には, この情報を適用するときユーザ入力として指示することを示す。また必須の空白は ~ で表す。例 2 に, 指定変数の度数分布を求めヒストグラム付きで出力するための SPSS の手続き制御カード列 SPEX 4 とそのジョブ制御カード列 SPJ 1 の記述例を示す。

例 2)

```

SPEX 4: RUN~NAME a; GET~FILE f;
FREQUENCIES vl, op='8'; FINISH;;
SPJ 1: SPSS f0='user. ident/..... pcc',
f4='user. ident/...Afile';
    
```

これらの制御カード列は, ShapeUp のワード列を

表 2 統計処理用対象プロセス
Table 2 Object processes for the statistical processing job in VCAP-1.

レベル名	対象プロセス名	説明
	O_V	VCAP のエントリプロセス
a	aO_P	プログラムパッケージ内の手続きの照会 ステップモードでのプログラムの実行 VCAP の支援によるプログラムの実行 V-file の管理
	aO_A	
	aO_H	
	aO_F	
b	bO_C	制御カードデッキの組み立て V-file から A-file へのデータ転送 統計パッケージのバッチ処理による実行 実行結果の出力
	bO_T	
	bO_E	
	bO_R	
c	cO_G	V-file の生成 V-file へのデータ入力 V-file のデータ出力 V-file の消去
	cO_S	
	cO_D	
	cO_R	

含むリスト形式で KBU に格納した。例 2 に示した SPEX 4 と SPJ 1 の格納例を例 3 に示す。

例 3)

```
IC ("SPEX 4", "SPJ 1",
  [ <<"RUN~NAME" "a">>, <<"GET~FILE"
    "f">>, <<"FREQUENCIES" "v1"
    "op='8' ">>, <<"FINISH">> ] ).
IJ ("SPJ 1",
  [ <<"SPSS" "f0='user. ident/... pcc'"
    "f4='user. ident/... Afile'">> ] ).
```

3.3 対象処理の分割と連携

統計処理における仕事の手順は SPSS と STAT PAC で共通であるので、2.1 節で述べた処理区分を反映して対象プロセスを設定し、その各々をさらに式 (4) の形式の処理片に細分して、VCAP-p のプログラムへ変換した。実際の処理を遂行するとき、ユーザの要求に応じて適用する処理片を選択し、メッセージの授受によってこれらを連結し実行するようにした。VCAP で設定した対象プロセスを表 2 に示す。

エントリプロセス O_V をつぎのように定式化した。

```
in [N] → [ShapeUp | Menu (L, N, O, E)]
  → out [O]
```

ここで用いる情報片 Menu の中の L は対象プロセスのレベル名、 N は通し番号、 O は対象プロセス名、 E はその説明である。 O_V の実行で出力する O を用いて後続の対象プロセスを選択する。

パッケージ内の手続きを照会するプロセス aO_P を

つぎのように定式化した。

```
in [k] → [ShapeUp | AR(k, n, p)] → out [n, p]
in [n] → [ShapeUp | AS(n, v, d, w, t_p, t_j, h)]
  → out [t_p]
in [n, p, t_p] → [COORD | CI]
  → out [n, p, TITLE, FUNC]
```

ここでは主に統計量をキーワード k として入力し、その該当するパッケージ名 n 、手続き名 p および表題 TITLE、機能 FUNC を出力する。遂行する処理は ShapeUp と COOD の連動により進めているが、この場合、ShapeUp が COOD の検索コマンドをファイル上に生成して COOD を起動する方式をとった。

他の対象プロセスも同様に定式化し、VCAP の内部で自動的に連携するようにした。たとえば、VCAP の支援を受けて統計パッケージを利用する場合は

```
 $O_V \rightarrow aO_H \rightarrow cO_G \rightarrow cO_S \rightarrow bO_C \rightarrow bO_T \rightarrow bO_E \rightarrow bO_R$ 
```

のように連携した。 bO_T の中でも、制御カード列を組み立てるために bO_C を用いている。

4. 統計パッケージ制御カード列の自動生成

統計パッケージを実行するための制御カード列を組み立てるための対象プロセス bO_C を次式の九つの処理片の連鎖で達成した。

```
in [k] → [ShapeUp | AR(k, n, p)] → out [n, p]
in [n, p] → [ShapeUp | AE(n, p, e, r_c)] → out [e, r_c]
in [n] → [ShapeUp | AS(n, v, d, w, t_p, t_j, h)]
  → out [t_p, t_j]
in [r_c] → [ShapeUp | IC(r_c, r_j, c)] → out [r_j, c]
in [t_p, c] → [COORD | CI] → out [FORM, PARM,
  EXPL, IFMT]
in [c, FORM, PARM, EXPL, IFMT] → [x | -] →
  → out [C_i]
in [r_j] → [ShapeUp | I_j(r_j, j)] → out [j]
in [t_j, j] → [COORD | CI] → out [FORM, PARM,
  EXPL, IFMT]
in [j, FORM, PARM, EXPL, IFMT] → [x | -] →
  out [J_i]
```

ここで使用する作用片としては既存の ShapeUp と COOD のほかに x を作成して用いた。 x は実用のプログラミング言語としては未熟である ShapeUp を補うためのもので FORTRAN で記述した。これは主に様式化した入出力やオペレーティングシステム内の他のモジュールとのインタフェースをとる部分である。 bO_C は、ユーザが入力するキーワード k を用い

表 3 AC 表および SC 表
Table 3 An example of table AC and SC.

ν	α	α	β	s	EXPL	IFMT
RUN~NAME	a	◇◇1		Frequencies of MBINCM	run 1	A64
GET~FILE	f	◇◇2		MB-file	file	A8
FREQUENCIES	v1	◇◇3		AGE,SEX,ED	varia	A
FINISH	op	◇◇4	++	8	opti	A

Table SC
SC(1)

FORM	EXPL
	FREQUENCIES \16 GENERAL=◇v1
	[OPTIONS \16 ◇op]
	[STATISTICS \16 ◇st]

SC(2)

PARM	IFMT	EXPL
v1	A	variable list or ALL
op	A	option list
st	A	statistics parameter list

て該当する使用例を捜し、それを参照してユーザ用の実際の手続き制御カードデッキ C_i とジョブ制御カードデッキ J_i を構築して出力する。各処理片が参照する情報は、AR, AE, AS, IC, CI (CIC), IJ, CI (CIJ) の順で、格納情報から一意に確定できない場合はユーザとの対話で決定する。

処理片 x は、取得した情報を用いて実際の制御カードデッキを組み立てるものである。他の処理片から x へのデータの受け渡しはファイル経由で行った。たとえば、リスト形式で格納した制御カード列 c や j を渡すときには、ワード列ごとに分解し、式(5)に示した ν の値につづいて式(6)のパラメータを

$$a, \alpha, 's', \beta$$

の形式のレコードに直して転送する。ただし、's' は r 項の文字列、 β は r 項の α である。一方、 c や j の中の ν の値を用いて CI から検索した対応制御カードの一般形式を、COOD のデータ出力形式 (項目名と値) のままファイル経由で x に送る。 x はこれらのデータを表 3 の AC 表と SC 表に再構成し、使用カードを順に解釈しながらつぎの手順で制御カードデッキを生成する。表を満たす情報の不足部分は、処理の途中で対話者に入力要求を出して指定してもらって補う。このときのプロンプト情報として CI から送られた EXPL の値 (パラメータの説明) を用いる。

① IC または IJ から送られたデータで AC 表を作成し、CI から送られたデータで SC 表を作成する。AC 表の EXPL, IFMT を SC(2) の値で埋める。

② AC 表の ν と a の値を用いて、SC(1) の FORM 中の不要綴りを除去し、制御カードデッキ CSET へ出力する。CSET にカードデッキの原形を作る。

③ α 欄につぎの規則で一意名を付け、CSET 中の対応するパラメータを書き換える。

- α 値が既指定ならそのままとする。
- α 値が未指定なら AC 表の行数を名前とする。
- 以上の後、名前に◇◇を付す。

④ s 欄が既指定なら β 欄に++マークを付す。

⑤ β 欄が空であれば AC 表の EXPL 欄を参照してユーザに質問を發し、そのとき入力された値を s 欄に書き入れる。表 3 に例示した s 欄の下線を付した部分が入力によって決定した値である。

⑥ β 欄と α 欄の名前を照合し s 欄を同じ値にする。

⑦ CSET 中のパラメータを s 欄の値で置き換える。パラメータの繰り返しがある場合は⑥に戻る。

⑧ CSET に作成されたカードデッキの位置付け指示に従って様式を整えた制御カード列を作成する。

以上の処理をまず手続き制御カードについて行い、つぎにジョブ制御カードについて行う。本来異なる形式の制御カードデッキの組み立てであるが、使用する情報片を入れ替えるだけで同じ処理片を用いて達成できた。図 4 に制御カードデッキを自動生成するための対話例を示す。

```

SYSTEM ?VCAP
VCAP's processing types, as follows :
1 ... Find procedures in program packages
2 ... Run program in step-mode
3 ... Run program with VCAP's assistance
4 ... Manage V-file
RETURN key terminates VCAP processing.
Which processing No.
? 2
Selected processing aOA ( Run program in step-mode )
Subprocessing types, as follows :
1 ... Construct actual control card deck
2 ... Transfer data from V-file to A-file
3 ... Execute application program
4 ... Output executing results
RETURN key terminates aOA processing.
Which aOA subprocessing No.
? 1
Selected processing bOC ( Construct actual control card deck )
Enter keyword
? frequency
SPEX3 ... SPSS FREQUENCIES
          frequencies of all variables
SPEX4 ... SPSS FREQUENCIES
          frequencies of indicated variables with histograms
Which control set ... (ex. SPEX3 ) ?
? SPEX4
run label (format A64) ?Frequencies of MBINCM
file name (format A8) ?MB-file
variable list or ALL (format A) ?AGE,SEX,ED

** Procedure Control Card
----- Actual control card deck -----
RUN NAME      Frequencies of MBINCM
GET FILE      MB-file
FREQUENCIES   GENERAL=AGE,SEX,ED
OPTIONS       8
FINISH

** Job Control Card
----- Actual control card deck -----
$ JOB
$ APROG  SPSS
$ PRMFL  I*,R,S,user.ident/.....pcc
$ PRMFL  FR,R,S,user.ident/...Afile
$ ENDJOB

Which aOA subprocessing No.
? 1
aOA terminated.

Which processing No.
? 2
VCAP terminated.

```

図4 統計パッケージ制御カード自動生成例

Fig. 4 An example of automatic construction of a control card deck for the statistical package (Underlined characters by the user's input).

5. 本方式の評価

VCAP は、ソフトウェア個々に依存する外部仕様を直接ユーザインタフェースとして使用せず、それらを制御情報データベース CI としてシステムに内蔵し内部で自動的に選択、解釈して利用するようにした。この結果、ユーザは同一の操作で異なるソフトウェアを利用することができる。また、これまで集約、共有されることの少なかったソフトウェアの利用に関する利用者の経験的知識を知識ベース KB に蓄積し、ユーザが適時参照できるようにするとともに、VCAP の実行中に VCAP 自身が内部で引用して自動的に処理を遂行するようにした。

事例として用いた二つの統計パッケージにおいては VCAP を介することにより、つぎのような利用面での改善をはかることができた。

(1) 本来異なる使用法をもつ SPSS と STATPAC を同一のユーザインタフェースで利用できるようになったので、ユーザは何分冊もの利用説明書 (参考文献 14) ~ 16) のほかに約 4 冊) を読む必要がない。

(2) 経験者が習得した利用上の知識を、VCAP の知識ベース KB を介して後進のユーザが利用できる。また、KB を更新することによって、新しい知識の追加や古い知識の削除をオンラインで行うことができるので、従来の印刷物による方法では困難であった有用な情報の伝達が容易に即時に行える。

(3) VCAP は KB に格納した知識を核にして駆動するので、ユーザ視野から統計処理に関連する各種ソフトウェア (統計パッケージ、データベースおよびファイル管理用ソフトウェア、バッチ処理依頼および結果受取り用システム) を体系化して提供することができる。また、KB を更新すれば VCAP の駆動も自動的に更新できるので、従来人手で行っていた情報提供が不要となる。

6. む す び

利用面からのソフトウェアの統合方式として知識中心のシステムを設計し開発した成果について述べた。本研究では、現在統計パッケージを実例としているが、この設計は、他のアプリケーションプログラムはもちろん、文献検索やマルチメディアデータベースの操作、分散型システムの制御指令など指令操作で実行するソフトウェアの統合にも適用可能である。また、これは、現在開発、利用の両面で問題となっているソフトウェアの危機を回避するために役立つであろう。今後は、知識の増加と管理、一層洗練した対話系、対象プロセスの拡充などについて研究を進展させるとともに、VCAP をインプリメントするために使用しているプログラム P, 特に知識情報処理用ツールをさらに最適なものに置き換えていく計画である。

謝辞 本研究は文部省科学研究費特定研究 (No. 58215001, 59209002) の補助により行った。関係各位に感謝する。

参 考 文 献

- 1) Bachman, C. W. and Williams, S. B.: A General Purpose Programming System for Random Access Memories, *Proc. FJCC*, pp. 411-422 (1964).
- 2) Roberts, L. G. and Wessler, B. D.: Computer Network Development to Achieve Resource Sharing, *Proc. SJCC*, pp. 543-549 (1970).
- 3) Smith, J. M. et al.: Multibase—Integrating Heterogeneous Distributed Database Systems, *Proc. NCC*, pp. 487-499 (1981).
- 4) 田中信行, 松田孝子: 統合学術情報システム TRACIS, 文部省科研費特定研究 TC 20Ⅲ-1-6, 東北大学 (1982).
- 5) 石田晴久ほか: 統合学術情報システム形成のための異機種間分散型データベースシステムの研究, 文部省科研費試験研究報告書, 東京大学 (1985).
- 6) Liskov, B.: On Linguistic Support for Distributed Programs, *IEEE Trans. Softw. Eng.*, Vol. SE-8, No. 3, pp. 203-210 (1982).
- 7) Haseman, W. D. and Whinston, A. B.: Automatic Application Program Interface to a Data Base, *Comput. J.*, Vol. 20, No. 3, pp. 222-226 (1977).
- 8) Warren, C.: Integrated Software Solves Complex Business Problems, *Mini-Micro Syst.*, Vol. 17, No. 6, pp. 195-202 (1984).
- 9) 溝口文雄: 知識工学, コンピュータソフトウェア, Vol. 1, No. 3, pp. 13-22 (1984).
- 10) 諏訪 基ほか: 人工知能とプログラミングの接点, 情報処理, Vol. 26, No. 5, pp. 521-529 (1985).
- 11) 米澤明憲: オブジェクト指向型プログラミングについて, コンピュータソフトウェア, Vol. 1, No. 1, pp. 29-41 (1984).
- 12) 松田孝子ほか: 関係情報データベース中心型統合ソフトウェア VCAP の一設計法, 情報処理学会アドバンストデータベースシンポジウム (1984).
- 13) 松田孝子ほか: 知識型統合ソフトウェア VCAP: 内蔵知識の適用による異種統計パッケージの制御, 情報処理学会知識工学と人工知能研究会, WGAI 42-10 (1985).
- 14) Nie, Norman, H. et al.: *SPSS Statistical Package for the Social Science*, second edition, McGraw-Hill, New York (1975).
- 15) Hadlai Hull, C. and Nie, Norman, H. (ed.): *SPSS Update 7-9 New Procedures and Facilities for Releases 7-9*, McGraw-Hill, New York (1981).
- 16) 統計解析システム説明書 STATPAC-6, 日本電気, DBX 02 (1982).
- 17) 松田孝子, 田中信行: ユーザ指向のデータベース管理システム COOD—設計とデータベース用言語, 情報処理学会論文誌, Vol. 21, No. 5, pp. 347-353 (1980).
- 18) ShapeUp 仕様書, 日本電気 C & C システム研究所 (1983).

(昭和 60 年 10 月 28 日受付)
(昭和 61 年 5 月 15 日採録)



松田 孝子 (正会員)

昭和 15 年生。昭和 38 年お茶の水女子大学理学部数学科卒業。理学博士。昭和 38 年三菱電機(株)入社。昭和 41 年東北大学電気通信研究所助手。昭和 44 年東北大学大型計算機センター助手。プログラム用言語, データベース用言語, ソフトウェアライブラリの研究開発に従事。最近情報システムにおけるソフトウェア構成論に興味をもつ。著書「JIS に準拠した FORTRAN」(共著, オーム社)など。日本ソフトウェア科学会会員。



鈴木 篤 (正会員)

昭和 12 年生。昭和 35 年米国ノックス大学経済学部卒業。昭和 40 年パーデュー大学経済学部大学院博士課程修了。Ph. D. 昭和 39 年インディアナ大学経済学部講師, 昭和 40 年同学部助教授, 昭和 41 年和歌山大学経済学部講師。

昭和42年同学部助教授，昭和48年東北大学応用情報学研究センター教授。計量経済学，経済時系列解析の研究に従事。統計データベースに興味をもつ。著書「統計学」(共著)。理論計量経済学会，日本統計学会各会員。



田中 信行 (正会員)

大正9年生。昭和16年12月東京大学理学部化学科卒業。理学博士。昭和22年東京大学助教授。昭和30年東北大学理学部教授。昭和52年大型計算機センター長(併任)。昭和58年退官。東北大学名誉教授。日本事務器(株)取締役会長。電極反応，無機溶液反応，情報処理など，最近はオフィスネットワークに興味をもつ。日本化学会賞受賞，紫綬褒章受章。日本化学会，アメリカ化学会等各会員。