

4. 高水準データベースシステムのための分析と評価

日本電気株式会社中央研究所 中田修二・真名垣昌夫

はじめに

EDPSの最近の動向として適用業務の拡大と統合化、広域化が進み、今後は利用の高度化、複雑化、処理や管理の分散化へEDPSが発展してゆくのは必然の姿であると思われる。この様なユーザの世界に近づくユーザ向けシステム (User Oriented System) [1] はデータベースシステム (DBS, Data Base System) を基盤として構築されると考えられる。

本報告ではDBMS (Data Base Management System) をよりユーザ指向へと拡張した高水準システムとしてアプリケーションベースシステム (ABS, Application Base System) を考え、このシステムの実現性をたしかめるためにDBSによる適用業務 (アプリケーション) の構造の変化を定量的に評価した結果について論じている。この結果、プログラム本数の減少やデータの重複度の減少などプロセス構造、データ構造の単純化の効果により高水準システムの要点である高水準インタフェース、アプリケーションの開発、導入、運用、保守、拡張等の容易化の見通しを得てABSの実現可能性がたしかめられた。

1. データベースシステム (DBS) の分析, 評価方法

DBSのシステム目的として従来より

- データ管理費用の低減化と変更費用の最小化
- データの利用可能性を容易に知り理解できる明快さ
- データの既存の使用のされ方を意識することなくデータベースの変更, 成長を容易にできる
独立性と柔軟性

などユーザインタフェースが高水準になると定性的に主張されてきたが [2], これらに対する定量的な評価は極めて数少ない [3]。我々は従来型ファイル (Conventional File) を利用したファイルシステム (File System) からデータ管理機能をシステム側にブラックボックス化する事によりDBSがどの程度ユーザにとっての使い易さを実現しているかを定量的に把握するためにアプリケーションのプロセス構造とデータ構造に着目して以下の様なパラメータを定めて定量的な評価を行なった。

(1) プロセス構造の評価パラメータ

- プログラム本数の減少
- アプリケーションの処理内容の変化
- サブシステム別のプロセス構造の変化
- プロセスフローの構造上の変化

(2) データ構造の評価パラメータ

- ファイル数とレコード, セット数の対応
- マスタレコードとトランザクションレコードの各々の特性によるレコード内の (論理的な) データ構造の変化
- データの重複度の減少効果
- 物理的なデータ量の変化

この様な項目について評価するときアプリケーションの特性を考える必要がある。このアプリケーションの特性としてDBSの分析に当たり以下に記す考え方により配慮した。

(1) アプリケーションタイプ : 情報の処理の仕方に注目して適用業務を以下の4タイプに分類した。

- 情報要約型, 原始情報処理型, 原始情報検索型, 情報即時処理型

(2) 処理プログラムの分類 : アプリケーションを構成するプログラムの処理機能は以下の10種

類に分類できる。

・入力変換，内部変換，分類，照合，併合，更新，抽出，分配，生成，作表

(3) サブシステムモデル：アプリケーションの設計法とユーザの処理プログラムの選択性に着目して，アプリケーションを以下に示すサブシステム構成にモデル化した。

- ・データ処理サブシステムは原始データの入力から，そのデータを用いた業務処理までの部分
- ・レポート作成サブシステムは処理されたデータから各種のレポートを出力する部分

(4) アプリケーションの比較基準：アプリケーションの設計開発者と利用者の各々の立場から見て分析，評価の主旨に合致するアプリケーションの対象範囲を明確にするため図1に示す様に以下の基準を定めた。

- ・開発形態 (M1) は開発者の用意したすべてのプログラムを含む場合
- ・最大利用形態 (U1) は利用者がアプリケーションを最大限に利用した場合
- ・標準利用形態 (U2) は開発者が用意したレポート作成サブシステムの1/2のプログラムをユーザが利用した場合
- ・自由作表形態 (U3) は開発者が用意したデータ処理サブシステムは利用するが，レポート作成サブシステムは利用しない場合
- ・自由入出力形態 (U4) はユーザ独自の入出力プログラムを用い，開発者が用意したアプリケーションからは業務処理の部分のみを利用する場合

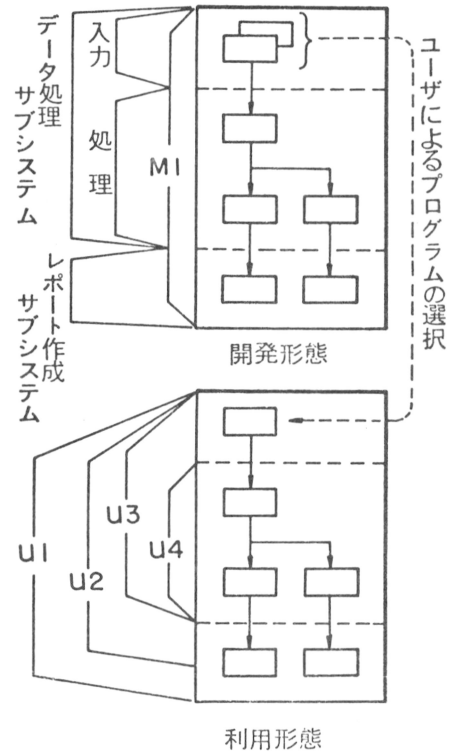


図1 アプリケーションの比較基準

II. データベースシステム化の分析，評価〔4〕

ファイルシステムからデータベースシステムへの移行を分析，評価する対象として当社のコンピュータシステム上で提供されている標準的なアプリケーションパッケージの中から販売管理と財務管理をとりあげ前述の標価パラメータについて評価を行なった。以下2.1と2.2で分析結果の一部を記述し，全体的な評価結果の要約を2.3に示す。

2.1 プロセス構造の変化の分析

(1) アプリケーションタイプ別分析：情報要約型の販売管理システムと原始情報処理型の財務管理システムのDBS化によるプログラム本数の減少をアプリケーションの比較基準に従って図2に表わす。この結果，以下の結論を得た。

一般にDBS化によるプログラム本数の減少効果はアプリケーションの特性（アプリケーションタイプ）により定量的には異なる。たとえばこの評価例では原始情報処理型（財務管理）は情報要約型（販売管理）に比べてプログラム本数の減少率が約15%程度高い。これは前者が個別情報を取り扱うための処理が情報要約型に比べて多く，この様な処理の多くがDBS化により不用となる事，さらにこの様にして不用となったプログラムの前後における更新処理を1つにまとめる事ができる場合がある事などによる。

(2) プロセス構造の変化：DBS化により削除可能なプログラムに注目し，そのプログラムのアプ

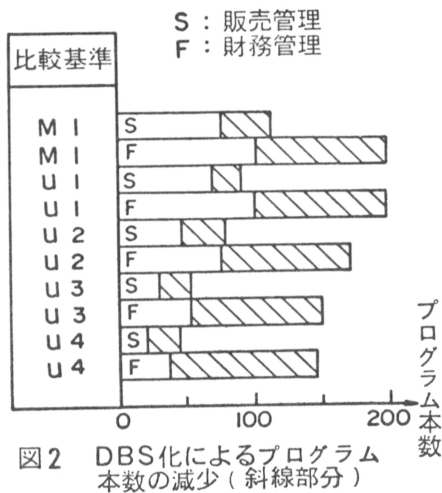


図2 DBS化によるプログラム本数の減少 (斜線部分)

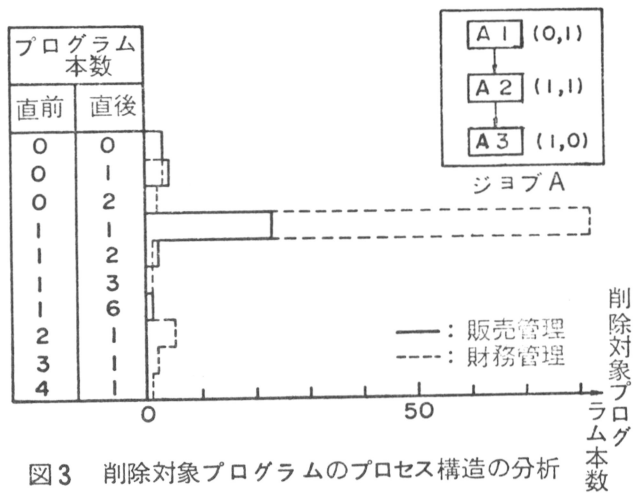


図3 削除対象プログラムのプロセス構造の分析

リケーションのジョブ構成、ジョブステップ構成の中での位置を調べた結果を図3に示す。ここでプログラムの位置は削除対象プログラムの直前、直後に実行されるプログラムの数により定めている。この結果、以下の結論を得た。

DBS化によるプロセス構造の変化は単純である。すなわち不必要となったプログラムの70%から80%が単純なシーケンシャル構造の中間に位置する。これらのプログラムの処理内容はその95%~100%が分類、内部変換、併合処理である。すなわち、これまでこれらの処理プログラムを含んで5~6個のプログラムで構成されていたジョブは3~4個のプログラムで行なえる様になりジョブを構成するジョブステップは短縮される。

2.2 データ構造の変化の分析

(1) DBS化によるデータの重複度の減少 : データベース化によるデータの重複度の減少をデータ項目件数について調べた結果を図4に示す。この結果、以下の結論を得た。

DBS化によりデータの重複は相当なくす事ができ、このためアプリケーションは理解しやすく、保守しやすくなる。データの重複度は一般にそのアプリケーションが対象としている業務の数や個別情報を取り扱う度合いが高い程高くなる傾向にある。レコードを識別するキィとなる様なデータ項目の重複が多い。

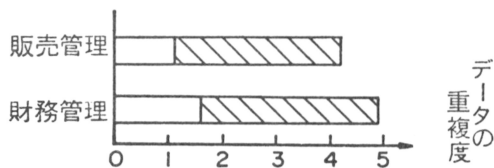


図4 DBS化によるデータの重複度の減少 (斜線部分が減少を示す)

(2) レコード内データ構造の簡単化 : データ構造の簡単化をより詳細に調べるためにレコードをマスターレコードとトランザクションレコードに2分して、レコードを構成するデータ項目の長さの総和により比較を行なった(図5)。この結果、以下の結論を得た。

マスターレコードの内容はほとんど変わらない。レコードのアイテム長の総和の減少率は約4%程度でアプリケーションタイプと無関係にほぼ一定している。減少しているのは主にキィとなるデータ項目である。トランザクションレコードはデータ項目の30%~50%が不必要となり、レコードのアイテム長の総和は30%~60%程度減少し、減少率はアプリケーションタイプに依存している。減少しているのは主に一般の(キィでない)データ項目である。

2.3 評価結果の要約

2.1と2.2でDBS化の定量的分析の一部を記述した。以下では紙幅の都合で詳述できなかった評

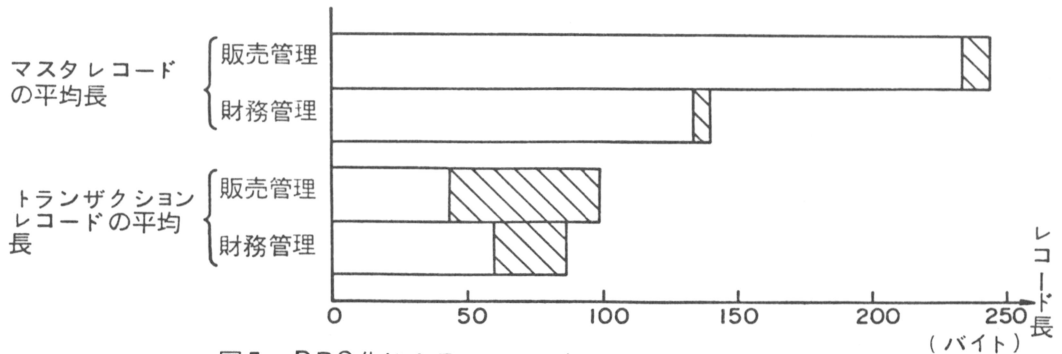


図5 DBS化によるレコード内データ構造の分析

(斜線部分がDBS化により削除されたデータ項目の長さの総和)

価パラメータに基づく分析結果の要約を記す。

2.3.1 プロセス構造の変化

(1) データベースを用いたアプリケーションの開発： プログラム本数は全体として30%から55%程度減少する。減少の程度はこの範囲内でアプリケーションの特性により異なる。

(2) プログラムの処理内容の面からの分析： 分類(ソート)プログラムはアプリケーションの特性に関係なく大巾になくす事ができる。内部変換, 併合プログラムはアプリケーションの特性に依存するが少なくとも60%以上はなくすことが可能である。更新処理は本質的には減少しない。ただし複数の更新処理や更新処理と他の処理を同一プログラムで行なえる様になる場合があり, この場合プログラム本数は減少する。

(3) サブシステム構成の面からの分析： データ処理サブシステムは50%から60%のプログラムが不用となる。これはこのサブシステムの約70%をしめる内部変換, 分類, 併合プログラムの多くが不用となる事による。一方, レポート出力サブシステムは10%から20%のプログラムが不用となる。これはこのサブシステムの約10%から20%を占める抽出, 分類プログラムが不用となる事による。

(4) プロセス構造の面からの分析： 削除できるプログラムの70%から80%が単純なシーケンシャル構造の中間に位置するプログラムである。その内の95%以上が分類, 内部変換, 併合プログラムである。これらのプログラムが削除される事によりジョブを構成するジョブステップが短縮する。この様にプロセス構造の変化は単純なので既存のプロセス構造をもとに容易にDBS化が計れる。

2.3.2 データ構造の変化

(1) データベースを用いたアプリケーションの開発： 従来型アプリケーションでデータの重複度が4.0から5.0程度のもものが, 1.0から2.5程度に減少する。

(2) レコード内の論理的なデータ構造からの分析： マスタレコードのデータ項目の構成はほとんど変わらず, データの減少率は約4%程度でアプリケーションの特性に関係なくほぼ一定している。この場合, 減少しているものは主にキィとなるデータ項目である。トランザクションレコードはそのデータ項目の30%から50%が不必要となる。この減少の程度はアプリケーションの特性により異なる。たとえば原始情報処理型の方が情報要約型よりデータの減少効果が大きい。この場合, 減少しているのは一般のデータ項目である。

(3) レコードの記憶構造(データ項目領域+ポインタ領域)からの分析： マスタレコードの物理レコード長は従来型ファイルのレコードより3%から10%程度長くなる。トランザクション

レコードの物理レコード長は従来型ファイルのレコードより10%から40%程度短くできる。

(4) 物理的なデータ量からの分析： 従来型アプリケーションの全ファイル量とデータベースとを比べれば、DB化によりほとんどの場合データ量は減少する。これは第1に従来型アプリケーションの入力データファイル、中間ワークファイル、出力イメージファイル等が不必要になる事による。第2に従来型アプリケーションのマスタレコード、トランザクションレコードのレコード長とレコード件数が与えられれば、これをDB化した場合のデータ量を概算できるが、一般にはデータ減少率の高いトランザクションレコードのレコード件数が多いのでデータは減少する事による。

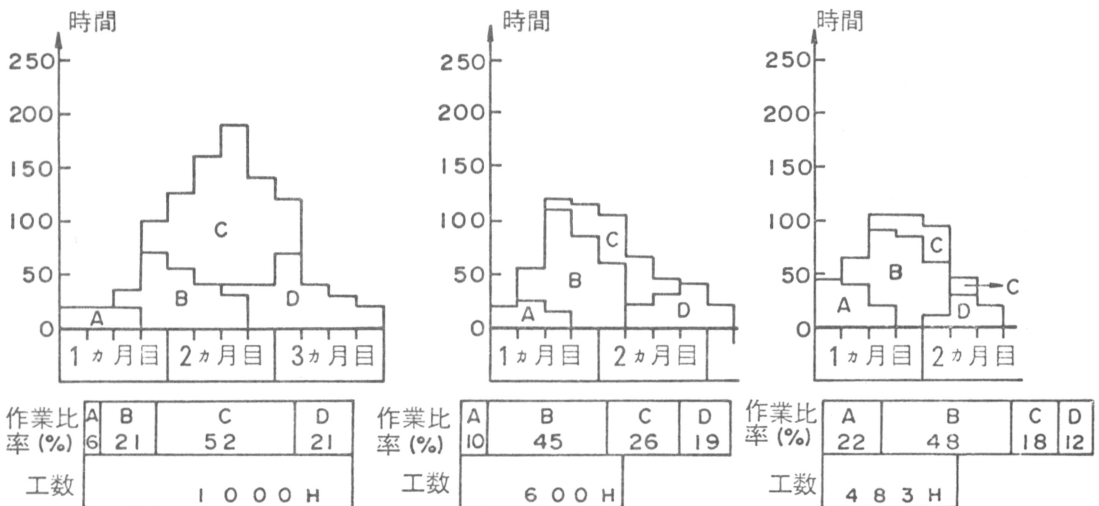
2.3.1, 2.3.2 で記した様にDBS化する事によりプロセス構造、データ構造共にわかり易くなり、使い易く、保守し易くなる。

2.4 DBS化の評価

2.3 までの結果を用いてDBSの有効性について考える。高水準システムの要点として高水準インタフェースとアプリケーションの設計開発の容易性が求められる。前者については前述の評価からプロセス構造の単純化にもとづき業務レベルのインタフェースが可能である事が判明した。以下ではアプリケーションの設計開発の容易性についての検討を記す。

従来よりソフトウェアの生産性向上を目的として構造化プログラミング手法やプログラムのモジュール化が行なわれ効果をあげている〔5,6〕。これに加えてDBSを利用することによる効果を評価した。その結果を図6に示す。DBSを用いる事により下記の効果が達成できる。

従来型ファイルを利用して構造化プログラミング等の手法を適用した場合に比べて70%から90%程度の工数でシステムの開発ができる。この場合、作業内容としてはシステム設計が2倍近くなり全体の20%から25%を占める。プログラム設計は全体の45%から50%を占め多少比率が高まっている。逆にプログラミング、テストは減少する。絶対工数で0.5倍から0.6倍でまかなえる。この作業内容の比率は各々約18%, 12%から13%程度になる。この様にデータ管理機能を持つDBMSを利用することにより、システム開発における負担の軽減とアプリケーションの設計により多くの力を集中しプログラミング以下の作業を容易にできる事が判明した。



(a) 従来型

(b) 従来型+IPT
手法

(c) DBS+IPT
手法

- A: システム設計
- B: プログラム設計
- C: プログラミング
- D: テスト & デバッグ

図6 アプリケーションの開発の評価

Ⅲ. アプリケーションベースシステム (ABS)

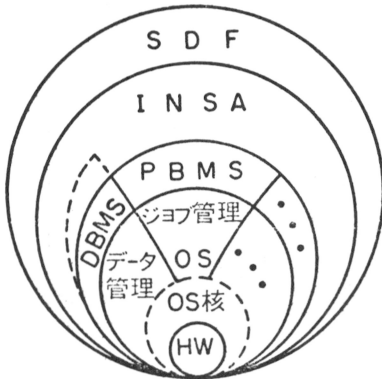
Ⅱにおいてアプリケーションシステムの設計, 開発, 稼働, 保守の面からDBSを基盤としてユーザにとって使い易い情報システムが構築可能であることがたしかめられた。ここでは高水準データベースシステムとしてアプリケーションベースシステム (ABS) の概要を記述する。

3.1 ABSとは

ABSはアプリケーションの(i)開発, 保守の容易性(ii)柔軟性, 種々のユーザ要求への適合性, (iii)利用の容易性(iv)変更, さらには拡張や移行の容易性を実現し, ユーザがEDPSの詳細を知らなくとも業務知識でシステムの利用を可能とする。このため以下の機能を実現する。

高水準システム機能の実現: ユーザの負担になっている機能をシステムで提供し, ユーザからはブラックボックスとして取り扱える様にする。ABSではユーザの適用業務から見て, データのみならず処理プロセスもアプリケーションの資源と見なせる点に着目してこれを管理するシステム機能を提供する。

高水準インタフェースの実現: システム内部でのアプリケーションの資源の管理とユーザが意識に持つ適用業務とのギャップを埋める, 使い易い業務レベルのインタフェース機能を提供する。



(a) アプリケーションベースシステム



(b) データベースシステム



(c) ファイルシステム

システム機能及びユーザインタフェース		システム	システム	システム	
		システム	システム	システム	
システム機能	アプリケーション開発支援			○	
	アプリケーション管理			○	
	プログラム管理			○	
	データ(統合)管理		○	○	
ユーザインタフェース	処理インタフェース	業務ユーザ		○	
		エンドユーザ		○	
		専門プログラマ	○	○	
	情報システム記述インタフェース	情報スキーマ	概念スキーマ		○
			分散型システムスキーマ		○
			ネットワークシステムスキーマ		○
		DBスキーマ	論理スキーマ		○
			物理スキーマ		○
			格納スキーマ		○
			探索スキーマ		○
		アプリケーションスキーマ		○	
ユーザプロセススキーマ		○			
ユーザプロフィール		○			

図7 システム構成の概要の比較

3.2 システム構成

ABSとデータベースシステム、ファイルシステムとの比較を図7、表1に示し、ABSの主要なシステム構成要素について以下で説明する。

情報システムアドミニストレータ (INSA Information System Administrator) はシステム情報を統括的に管理しユーザとのインタフェースに関する様な高レベルのアプリケーション

システム目標	高水準システム		高水準インタフェース	
	データベース管理システム (DBMS)	プログラムデータベース管理システム (PBMS)	情報システムアドミニストレータ (INSA)	(超) 高級言語
システム完全性	<ul style="list-style-type: none"> データ内容の完全性の維持 	<ul style="list-style-type: none"> プログラム内容の完全性の維持 プログラム間の関係、構造の維持 	<ul style="list-style-type: none"> システム情報の統括管理によるユーザ、プロセス、データの完全性の維持と管理 	
システムの使い易さ	<ul style="list-style-type: none"> データの統合管理 アプリケーションマによるデータの仮想化 エンドユーザ言語 複数のユーザによるデータの利用の競合管理 データ間の関係による即時処理 	<ul style="list-style-type: none"> プログラムの追加、変更、消去等の支援 	<ul style="list-style-type: none"> ユーザ、プロセス、データのデクシヨナリ、ディレクトリ プロセス構造、情報構造の仮想化 	<ul style="list-style-type: none"> 業務ユーザインタフェースなどのユーザのEDPスキルに合った処理言語
利用の容易性				
可用性の増大			<ul style="list-style-type: none"> 分散型データベースの管理と仮想化 非定型処理の支援 システム機能縮退の支援 	
開発の容易性	<ul style="list-style-type: none"> 処理プログラム、データ構造の明快さによる変更の容易性 	<ul style="list-style-type: none"> プログラムの内容構造、相互関係などの追加、変更、消去等の支援 	<ul style="list-style-type: none"> 分散型構造のカタログ化による仮想化、処理のプログラクホックス化 	<ul style="list-style-type: none"> 設計開発言語の提供 i) 分散型情報システム記述言語 ii) カスタマイズ言語 など
ユーザへの適合性				<ul style="list-style-type: none"> 情報モデル、EDPモデルの各レベルからの開発支援 ユーザシステムの設計評価 (シミュレータ) など
移動、変更、保守の容易性				<ul style="list-style-type: none"> 多種多様なアプリケーションの部分的導入、追加登録の支援
拡張、移行の容易性			<ul style="list-style-type: none"> 分散型システムで新しいノード、データベース等の追加のためのデータ生成 	

表2 アプリケーションベンチマークのシステム機能

ョン資源の管理を行ない、ユーザのシステムの利用とシステム内の他の構成要素を支援する。INS Aはシステムの広域化による利用や管理の複雑さを避けるため特に分散型の情報システム〔7, 8〕では重視される。

システムデザインファクトリ(SDF System Design Factory)はアプリケーションの設計, 開発, 顧客化, 導入の支援ツールである。DBSの環境においてはアプリケーションの設計の重要性が増加し, プログラミングやテストの比率が減少する事が明らかとなったが, SDFはABSの上でのアプリケーションの設計, 開発を主対象に支援を行なう。

プログラムベース管理システム(PBMS, Program Base Management System)はINS Aの統括的な制御下においてシステムプログラム, ユーザプログラムなどをソース形式, オブジェクト形式で管理し, 業務の変更等によるプログラムの変更, 保守等の作業を容易化する支援機能を提供する。

3.3 システム機能

ABSのシステムとしての目標を達成するために, ABSの主要な構成要素が提供する機能を表2に示す。

IV. おわりに

本報告ではアプリケーションベースシステム(ABS)をデータベースシステム(DBS)を基盤として実現するための定量的評価を行なった。この結果としてここでは, DBS化によりプログラム本数が30%から55%程度減少し, データの重複度も4.0から5.0の範囲であったものが, 1.0から2.0程度に減少する事が明らかになった。さらにこの様なプロセス構造, データ構造の変化によりアプリケーションがよりユーザの意識に近い本質的な処理により構成される様になる事が明らかになりDBS化の有効性が確認できた。この結果DBSをよりユーザ指向へと拡張した高水準情報システムであるABSの実現可能性を確認し, そのシステム概要を紹介した。

最後に, 本研究の機会を与えてくださった当コンピュータシステム研究部藤野部長, 日頃御指導をいただいている三上課長に感謝の意を表わします。

参考文献

- [1] 国井: 高水準データ・ベースシステムの現況と将来, 情報処理学会・データベース研究会資料(1976, 6)
- [2] Martin, J.,: Computer Data - Base Organization, Prentice Hall (1975)
- [3] 窪田: データベースの実施例について, 情報処理学会・データベース研究会資料(1974, 9)
- [4] 中田, 真名垣: データベースシステム化によるアプリケーションの構造評価, 情報処理学会第17回全国大会
- [5] 中原: ソフトウェア開発技術の現状と問題点, ビジネス・コミュニケーション, Vol 13, No 5, pp.26~31 (1976)
- [6] 篠沢, 池田, 渡辺: 論理/物理プログラム設計論, 電子通信学会・電子計算機研究会資料(1976, 2)
- [7] 真名垣, 中田: データベースシステムアドミニストレーションシステム, 情報処理学会第1.6回全国大会(1975)
- [8] 真名垣: 分散型データベースシステムの制御, 情報処理学会第17回全国大会(1976)



本 PDF ファイルは 1977 年発行の「第 18 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場（＝情報処理学会電子図書館）で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>