

TECHNOLOGY

REVIEW

UNISYS

技 報

通巻
20

1989年2月発刊

Vol. 8 No. 4

論 文

AIS 1100 の分散データ・アクセス	福島康夫, 米津政紀	1
ESD におけるエネルギー変換とその妨害作用	本田昌實	27
LINC 適用によるソフトウェア開発事例	平山道彦	42
順序ソート代数に基づく代数的仕様と 始代数存在判定	清水敏彦	57
H-CAD とプレカット工法による 一貫ハウジング・システム	横井時人	75
鋳物凝固解析プログラム CAST の アルミニウム合金鋳物への適用	松林幸枝	95

技術動向

OSI の現状と動向	當麻悦三	105
------------------	------	-----

図書紹介

掲載論文梗概		表 2
--------------	--	-----

分散処理システムが一般化しつつある。当社のユーザにおいても、大規模システム・ユーザでは単一集中型ホスト・システムから、拡張性・柔軟性を重視した分散型ホスト・システムへの進化や、企業組織形態に合わせた分散型情報処理システムの構築に向かうユーザ等分散処理システム化が浸透している。福島康夫、米津政紀のAIS 1100の分散データ・アクセスは、金融第三次オンラインの分散データ資源のアクセスに関するニーズに基づき、AIS 1100*の環境下での実現方式について、とくに機能面を中心として言及している。

*AIS 1100: Advanced Information System for UNISYS series 1100

帯電物体に蓄えられていた静電エネルギーは、スパーク放電によって、熱、光、イオン化そして電磁(界)エネルギー等に変換される。とくに静電気放電(ESD)の熱的な作用に関しては可燃性気体、液体、粉体等を扱っている産業においては爆発事故等の致命的な災害の原因になるため、従来から多くの研究が成されてきた。本田昌實のESDにおけるエネルギー変換とその妨害作用は、空気中における金属物体間の静電スパーク放電による放電前後のエネルギーの変化と、金属物体の近傍に生ずる電磁界の性質についての実験報告である。

第四代言語 LINC の適用は、少い工数で高品質なシステムの開発を可能にしている。しかし、その効果のほとんどはソフトウェアの開発工程の下流部分(プログラム開発工程等)におけるものである。平山道彦は、LINC 適用によるソフトウェア開発事例の中で、開発工程の上流(分析・要求定義・設計)部分での高品質・高生産性の実現を目指し、開発工程の上流はデータフロー・ダイアグラム(DFD)を利用して要求モデルを作成し、下流はそのDFDによって作成された要求モデルをインプットとしてLINCによりプログラムを生成する手法を適用した総合情報システムの開発状況を報告している。

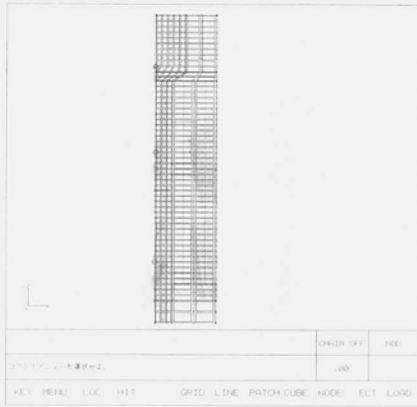
順序ソート代数に基づく代数仕様は、従来の多ソート代数に基づく代数的仕様記述法を、ソート

が示す集合の間に包含関係を定義できるように拡張したものである。この方法では、従来の多ソート代数に基づく方法では記述できなかった、集合間の包含関係、部分演算、演算の多種継承、多義演算を表せ記述能力が高い。しかし、ソート間の半順序が代数に新しい制約をおき、記述した仕様の意味を定める始代数が存在しないことが起こる。清水敏彦は、順序ソート代数に基づく代数的仕様と始代数存在判定の中で、記述した仕様の始代数の存在を判定するために、存在の十分条件である正則性と接続性を調べる手続きを実現したことを報告している。

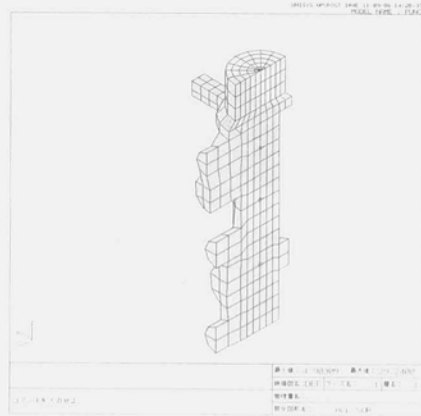
住宅業界は、活況な中で有数企業への集約化、工場での生産加工化が進み、これに伴ないコンピュータへの関心も高まりつつある。すでに開発されて5年になる住宅CADシステム(H-CAD)は、住宅業務の流れに沿って情報が作られ、出力できる一貫システムを特長としている。H-CADのデータベースはコンピュータ内部に、屋根・外壁などを「部位」展開し、窓・流し台などを「配置」することにより、3次元の家モデルを忠実に作り上げるといった考えを持っている。横井時人は、H-CADとプレカット工法による一貫ハウジング・システムの中で、H-CADおよび検索システムを始めとする周辺サブシステムが各業務の中でいかに適合されるかについて述べ、さらにそれらのシステム構成、データ構造についても説明を加えている。

鋳物製品に対するコスト低減と品質の向上、特に鋳造欠陥防止への要求が強く、解析技術の利用によるひけ巣予知法が近年多く試みられている。松林幸枝は、鋳物凝固解析プログラムCAST*のアルミニウム合金鋳物への適用の中で、凝固収縮流の処理をも考慮に入れた有限要素法による3次元鋳物凝固解析プログラムCASTを紹介し、これまで機会がなかったアルミニウム合金鋳物の凝固過程をCASTにより分析し、実験結果との比較検討を行ったことを報告している。

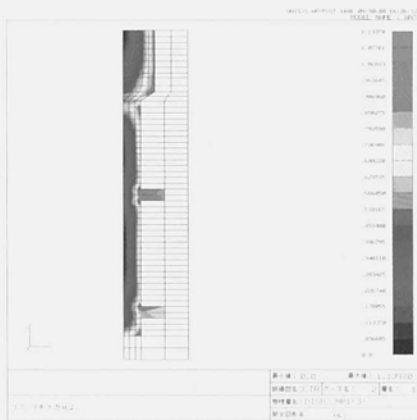
*CAST: CASTing analysis program



口絵 1 回転軸対称体の形状



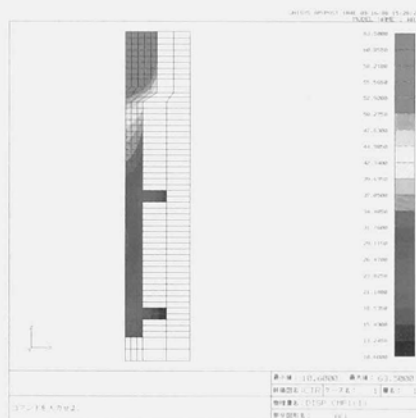
口絵 2 プレーキシリンダの形状



口絵 3 モデル 1 G/\sqrt{R} 図



口絵 4 モデル 1
ひげ巣のカラーチェック

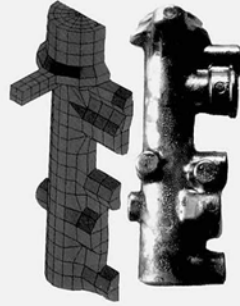


口絵 5 モデル 2 等凝固時間図

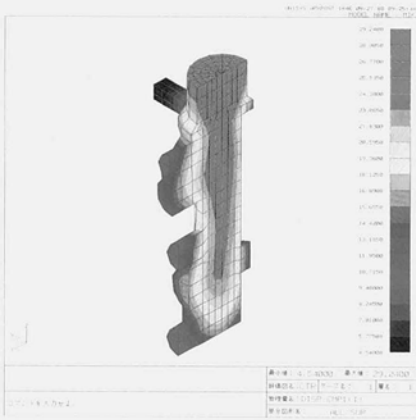
詳細は、本文「鋳物凝固解析プログラムCASTのアルミニウム合金鋳物への適用」に掲載。



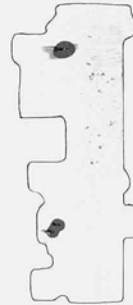
口絵6 モデル3 実物図



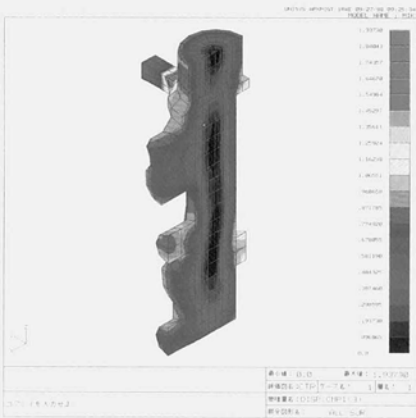
口絵7 モデル3 メッシュ分割図



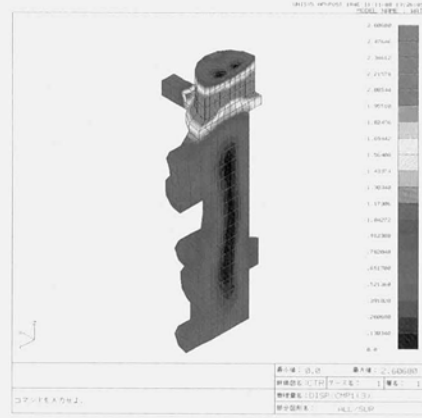
口絵8 モデル3 等凝固時間図



口絵9 モデル3
ひけ巣のカラーチェック



口絵10 モデル3 G/\sqrt{R} 図



口絵11 モデル3 G/\sqrt{R} 図
(最上部の熱容量10倍)

詳細は、本文「鋳物凝固解析プログラムCASTのアルミニウム合金鋳物への適用」に掲載。

AIS1100 の分散データ・アクセス

Remote Data Access on AIS 1100

福島康夫, 米津政紀

要約 分散処理が一般化しつつあり, 日本ユニシス (NUL) のユーザにおいても分散化指向の高まりが見られる。

本稿は金融第三次オンラインの分散データ資源のアクセスに関するニーズに基づき, AIS 1100 (Advanced Information System for UNISYS series 1100) の環境下での実現方式について, とくに機能面を中心として言及している。

該機能は, AIS 1100 の分散処理支援のためのサブシステム ASCOT 1100 (AIS System CONNecTion for UNISYS series 1100) の選択機能 RDA (Remote Data Access) により提供される。

そのうちの主な機能は次の通りである。

- 1) 業務処理に対する分散不可視性 (分散を意識させない外部インタフェース) の提供
- 2) 2 フェーズ・コミットメントによるデータ保全性の保証
- 3) ネットワーク・ワイドのデッドロックの検出
- 4) 複写型データベースの維持・管理

Abstract The need of distributed data processing is increasing among Nihon Unisys (NUL) customers in line with the on-going generalization of remote data access.

This paper mainly describes the AIS 1100 features for establishing the remote resource access method used for 3rd -phase financial online systems.

These features become available by adopting ASCOT 1100/RDA (Remote Data Access) which is an optional subproduct of AIS 1100.

The main features are:

- 1) to provide location transparency
- 2) for two-phase commitment
- 3) for network-wide dead lock detection
- 4) to support the replicated database

1. はじめに

分散処理システムは最近のさまざまな技術的進歩に支えられ, 一般化しつつあるといえる。当社のユーザにおいても, 大規模システムのユーザでは単一集中型ホスト・システムから, 拡張性・柔軟性を重視した分散型ホスト・システムへの進化や, 企業組織形態に合わせた分散型情報処理システムの構築に向かうユーザ等, 分散処理システム化が浸透している。

統合オンライン・ソフトウェア AIS 1100 (図1) もこれらの要請に応えるべく, 分散処理機能の拡充を図っており, AIS 1100 のサブ・システム ASCOT 1100 (図2) としてプロダクトの提供を開始しつつある。

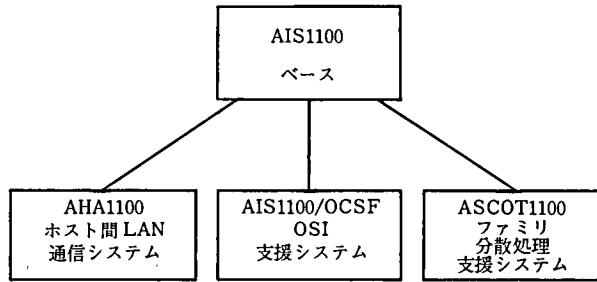


図1 AIS1100 ファミリー・ソフトウェア・プロダクト

Fig.1 AIS1100 family software product

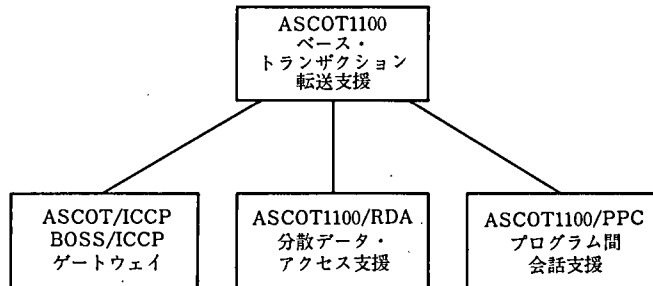


図2 ASCOT1100 ファミリー・ソフトウェア・プロダクト

Fig.2 ASCOT1100 family software product

本稿は、ASCOT 1100 の選択機能の一つである RDA (Remote Data Access ; 以下 ASCOT 1100/RDA) について、機能および実現方式を報告する。

1.1 分散データ・アクセス機能開発の背景

分散データ・アクセス機能の必要性は、直接には金融第三次オンライン・システム構想において要請されている。この構想における中心課題は、1990年代における業務量の飛躍的増加に対応すべく業務ホストの複数化による分散処理を基本としたシステム基盤の構築である。具体的には各業務を地区分割し、センタ内の複数ホストに各々処理を分散させるシステムの実現である。分割数と分散ホスト数は、容易に拡張可能であることが必要である。

このような分散システムでは、とくにデータの分散とアクセスの方式がそのシステムの拡張性や効率といった重要な目標に対して影響を与える。いかに他との関連性が少なく、独立したサブセットにデータを分割できるかが鍵となるが、完全に独立したサブセットへの分割は極めて困難である。このような場合、一つのトランザクション処理においてどのように分散データをアクセスするか、その方式を確立することが必須である。この方式には大きく次の二つがある。

- 1) トランザクション静止型……最初にトランザクションを受け取ったホストで業務処理を行う。業務処理対象データはそのホストに送られ、更新データが返送される。通信量は、データ静止型に比較して多く実現機構も複雑であるが、極めて柔軟性に富む。

2) データ静止型……業務処理は業務処理対象データの存在するホストで行う。最初にトランザクションを受け取ったホストでは、業務処理対象データのあるホストに処理を依頼するためトランザクションを転送する。通信量は少ないが、業務処理レベルでの考慮が必要となり柔軟性に欠ける。

金融第三次オンライン・システムでは、業務間連動や他店顧客処理においてこのような分散データ・アクセスが発生する。当該システムでは、つぎに述べる理由により‘トランザクション静止型’の対応が要求されている。

- ① 多くのアプリケーション・プログラム (以下 AP) が既存プログラムからの移行であり、データ静止型を採用するとプログラム構造の大幅変更となる。
- ② 分割数、分散ホスト数を自由に拡張できるように、AP はアクセスするデータがどのホストに存在するかを意識しないようにしたい (分散不可視性の獲得)。
- ③ 分散データ・アクセスが必要となるトランザクション量は、相対的にごく少なく全トランザクション量の 1~3 %程度と見積もられる。

なお、‘データ静止型’の対応は、すでに ASCOT 1100 1R においてホスト間通信機能としてリリースされている。

1.2 プロダクトの目標および目的

ASCOT 1100/RDA は、通信媒体で結合された遠隔地にあるホスト・コンピュータのデータに対し、AP からの自由なアクセス参照、検索、更新を可能とする。AP は、アクセスすべきデータがどのホスト・コンピュータ上に存在するのか意識する必要はなく、またデータおよびそのデータに対するアクセスと処理について発生しうる障害において、データの安全性が保証される。

ASCOT 1100/RDA の機能目標は次の通りである。

- 1) 業務処理に対する分散を意識させない外部インタフェース (分散不可視性)
- 2) 2 フェーズ・コミットメント・プロトコルに基づく、データ安全性の保証
- 3) ネットワーク・ワイドのデッドロック検出
- 4) 要求および結果の転送
- 5) ネットワーク・ワイドなデータ・ディレクトリの維持・管理
- 6) リクエスト/サーバの提供
- 7) 複写データの維持・管理
- 8) 通信媒体一回線・IPCC・HYPER channel への対応
- 9) 集中型データとの共存および統合
- 10) 分散対象データの種類の依存しない汎用的機能

これらを実現することにより、以下を可能とすることが ASCOT 1100/RDA の目的である。

- ① 負荷分散による大幅な処理効率の向上
- ② 障害の局所化による高信頼性の確保
- ③ 柔構造システム化によるシステム拡張性の確保

2. 分散データ・アクセスの実現方式

本章では AIS 1100 のサブ・システムである ASCOT 1100/RDA の分散データ・ア

セスの実現方式について記述する。

2.1 AIS 1100 の概要

AIS 1100 は、オンライン機能(DC)とデータベース機能(DB)を有機的に結合し、開発から運用に至るシステム・ライフサイクル全般を支援する統合オンライン・ソフトウェアである。

AIS 1100 は、処理内容の高度化や取り扱う情報の多様化に対応するため、より高度な機能を標準的に提供し、システム構築の各局面での使用者負荷を削減し、業務処理に専念できるようなシステム環境を実現している (図3)。

AIS 1100 の機能概要は次の通りである。

- 1) 統合回復機能……OS 1100 (UNISYS シリーズ 1100/2200 の OS) の統合回復機能 (IR: Integrated Recovery) により、すべてのデータベース・メッセージにかかわるシステムの各種障害に対し統一的な手順で回復する。
- 2) オンライン機能……メッセージ保存やリカバリ、通番管理、再送など豊富なメッセージ制御機能を提供する。
- 3) データベース機能……DMS 1100 (Database Management System 1100) の提供するネットワーク・データベース, RDMS 1100 によるリレーショナル・データベースを容易にアクセスできる。
- 4) AP 支援機能……ディレード (遅延)・バッチ処理, センタカット処理, システム・テーブル, タイマ起動などの機能を提供する。
- 5) 開発支援機能……AP は, AIS 1100 の提供する単純で高水準なインタフェース

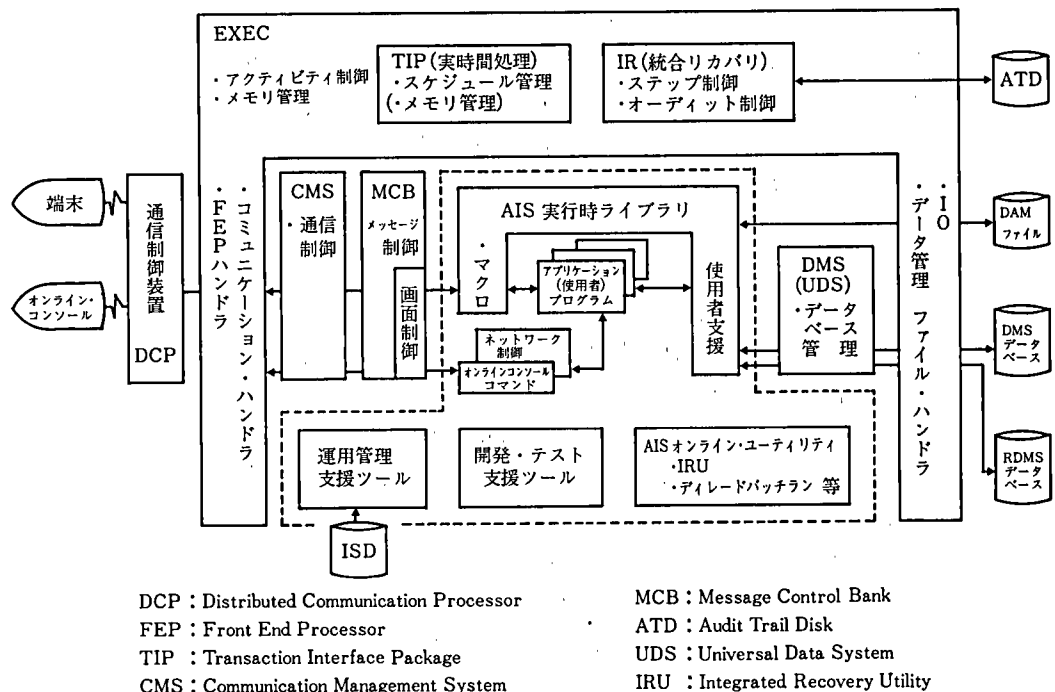


図3 AIS1100 概要図

Fig. 3 The outline of AIS1100

を使用することにより、リカバリやエラー処理など複雑なオンライン制御を意識することなく、業務処理に専念することができる。このインタフェースでは、リカバリ準備、I/Oエラー、デッドロック・エラー、プログラム異常終了などの処理はすべてブラック・ボックス化されており、使用者はとくに意識する必要はない。

また、開発・テストを支援するための機能としてトレース、パッチ端末シミュレータ、デマンド端末シミュレータ、ネットワーク・シミュレータを提供する。

- 6) 運用支援機能……統合システム・ディクショナリ (ISD: Integrated System Dictionary) によりシステム構成情報 (ネットワーク構成, データベース・スキーマ, システム動作環境等) を一元管理し, 必要なシステム・パラメータ, システム開始・終了・再開 JCL を自動生成する。また, システムに対する動作指示, 状態監視のためのオンライン・コンソール, 標準コマンドを提供する。

2.2 ASCOT 1100/RDA による分散データ・アクセス

ASCOT 1100/RDA は, 通信ネットワークにより結合されたコンピュータ・システム上に存在するデータ (データベース, ファイル, システム・テーブル) をアクセスする高水準な論理インタフェースを提供する。AP は ASCOT 1100/RDA により, データの物理的所在, ネットワーク等の物理構成, リカバリ等を意識することなく, 分散配置されたデータへのアクセスが可能となる (図 4)。

2.2.1 データの分散単位

分散されるデータが所属し, 管理される一つの単位をノードと呼ぶ。ノードは, ネットワークワイドに識別可能なアドレスを持つ。ASCOT 1100/RDA では, ノードはアプリケーション・グループ (OS 1100 の Integrated Recovery の単位) に対応する。

データの分散単位は AP が処理可能な論理的な単位で設定する必要があり, ファイルおよびテーブルではレコードが該当する。データベースでは, 分散単位は論理的に

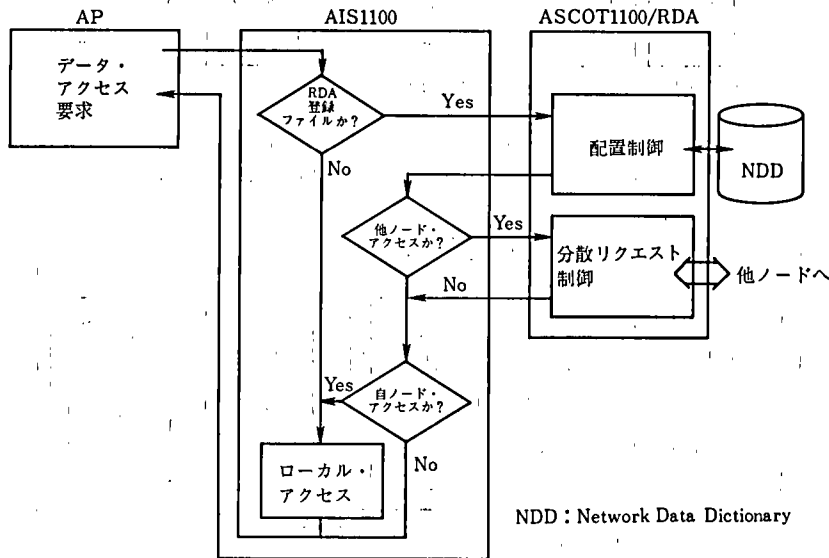


図 4 プログラム・インタフェースによる分散不可視性
 Fig. 4 The location transparency of program interface ,

は DBTG のセット・タイプとし、物理的にはセット・オカレンスとエリアとする。

なお、データベース処理要求のノード内完結性を高めるため、セット・オカレンスは一つのエリアに閉じて格納し、一つのエリアは唯一のノードに配置するものとする。

2.2.2 機能構成

ASCOT 1100/RDA の機能構成およびソフトウェア構成は図 5、図 7 の通りである。配置制御機能は AP からの要求を受け取り、どのノード中のデータにアクセスするかを、NDD により決定する。

分散アクセス制御機能は、ユーザ要求が他ノード中のデータへのアクセスを伴う時に、他ノードへのアクセス依頼と、アクセスの応答を制御したり、他ノードにわたるデッドロックを検出する。ASCOT 1100/RDA は、分散アクセスをいわゆるファンクション・ SHIPPING により実現する(図 6, 7)。更新同期制御・リカバリ機能は、複数ノードにわたる更新処理要求について障害に対するデータの一貫性を保証する。

通信インタフェースは他のノードとの通信を物理媒体(回線, チャネル, ハイパー・チャネル等)と独立に行うためのインタフェースを提供する。

NDD は分散システムを構成する各ノード中のデータの物理的配置, データの属性, アクセス・ロジック等を定めたものである。

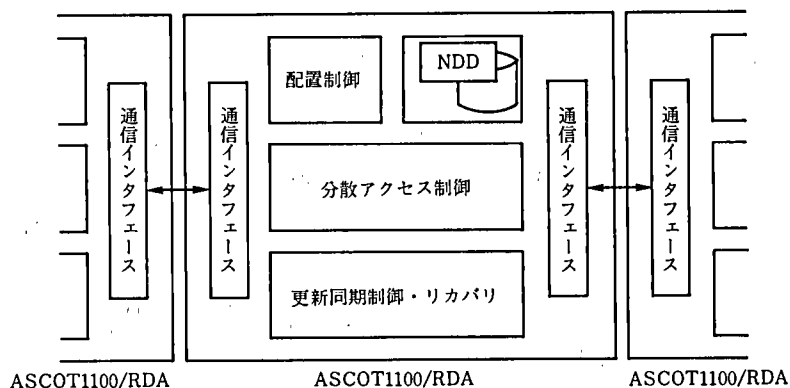
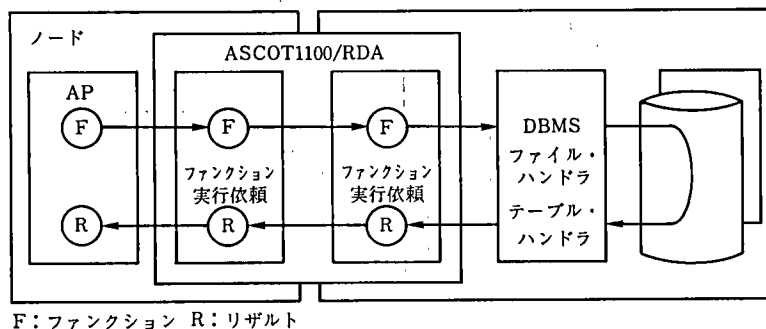


図 5 ASCOT1100/RDA の機能構成

Fig. 5 Function components of ASCOT1100/RDA



F: ファンクション R: リザルト

図 6 ファンクション・ SHIPPING

Fig. 6 Function shipping

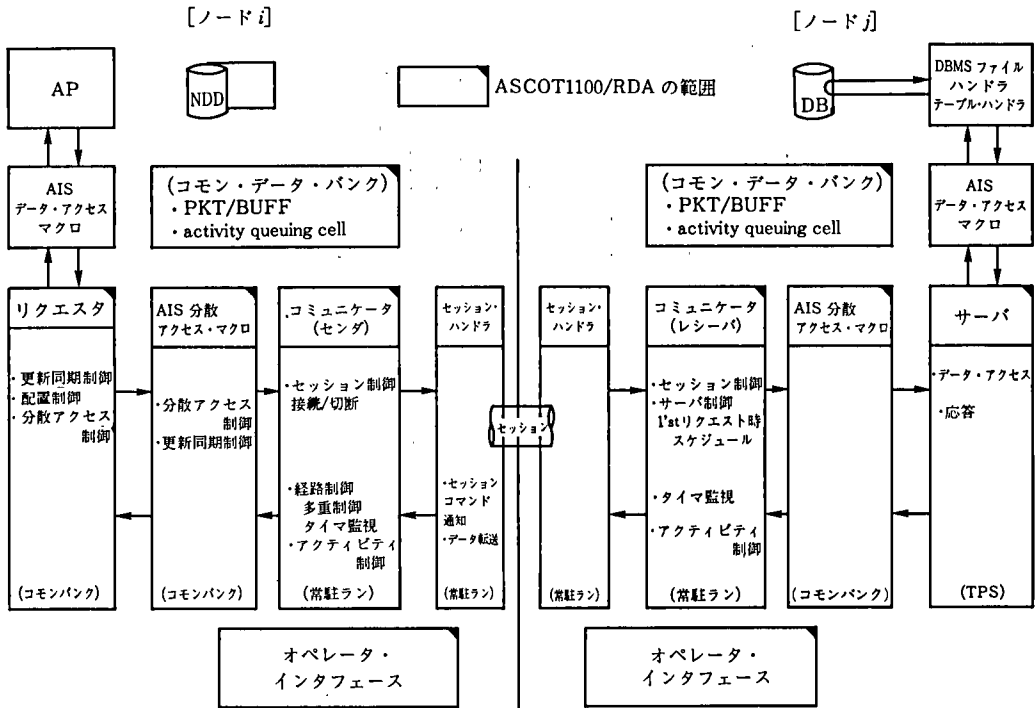


図7 ASCOT1100/RDAのソフトウェア構成

Fig. 7 The software configuration of ASCOT1100/RDA

3. 配置制御

分散システムでAPからデータ・アクセスが要求された場合、最初に必要なのはデータ所在の認知である。配置制御とは、処理に必要なデータの存在するノードおよび該当ノード内のデータ配置を決定する機能である。

配置制御機能はAPからのデータ・アクセス要求を受け取り、どのノードにアクセスするかをNDD, CRT (CuRency Table) により決定する。

3.1 分散データの配置形態

ASCOT 1100/RDAは、AIS 1100で提供しているデータベースに対し次の三つのファイル・タイプを提供する (図8, 9, 10)。

- 1) 分割ファイル……同一タイプのレコードが物理的に複数のファイルに分割されており、ユーザは論理的に1個のファイルと認識している。DMS 1100のスキーマ定義にあるマルチ・エリアの概念がこれに該当する。この場合、あるレコードはシステムに一つしか存在せず、あるホーム・ノードに割り当てられている。レコードを、それに対するアクセスが最も頻繁に行われるノードに割り当てて応答時間と通信量を減少させる。このタイプの一つの鍵となる性質は、ローカルでないレコードのアクセス頻度である。
- 2) 複写ファイル……分散DB環境内に同一のファイルが複数個存在し、それぞれ別のノードに存在する。処理能力や信頼性等の改善が図られるが、このタイプで

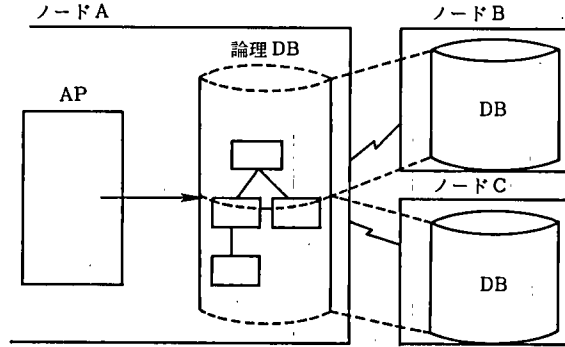


図8 分割型データベースの例
Fig. 8 Partitioned type

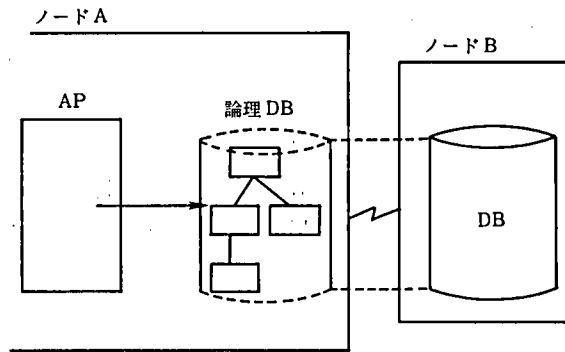


図9 複製型データベースの例
Fig. 9 Replicated type

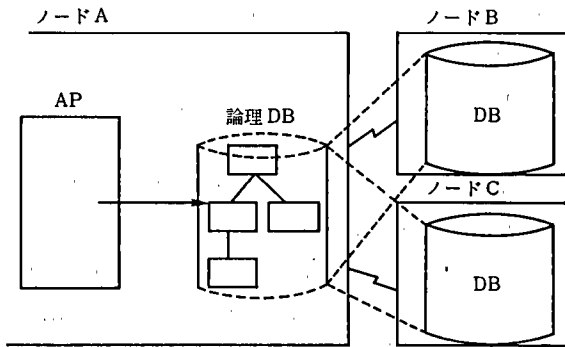


図10 固有型データベースの例
Fig. 10 Native type

は更新の頻度が鍵となる。

- 3) 固有ファイル……分割型が、論理的に一つのデータベースとみなせるデータを物理的に複数のノードに分散させるのに対し、元来アプリケーション・ネットワーク内にあるアプリケーション・システムにローカルに存在するデータを固有型という。分散アクセスの観点からは分割型と変わりはない。

3.2 分散データ管理

分散データベース・システムを構築するために1ノード内のスキーマ概念を拡張し(図11), DMS 1100により管理されているラン・ユニット単位のカレンシ制御も分散システムで管理する。

また、データベースと同様に、OS 1100/TIP (Transaction Interface Package: シリーズ 1100/2200における汎用リアルタイム・パッケージ) 特有のファイル (FCSS: File Control SuperStructure)/システム・テーブルも提供しており、FCSS ファイル番号+レコード番号、テーブル番号+レコード番号で配置を管理する。

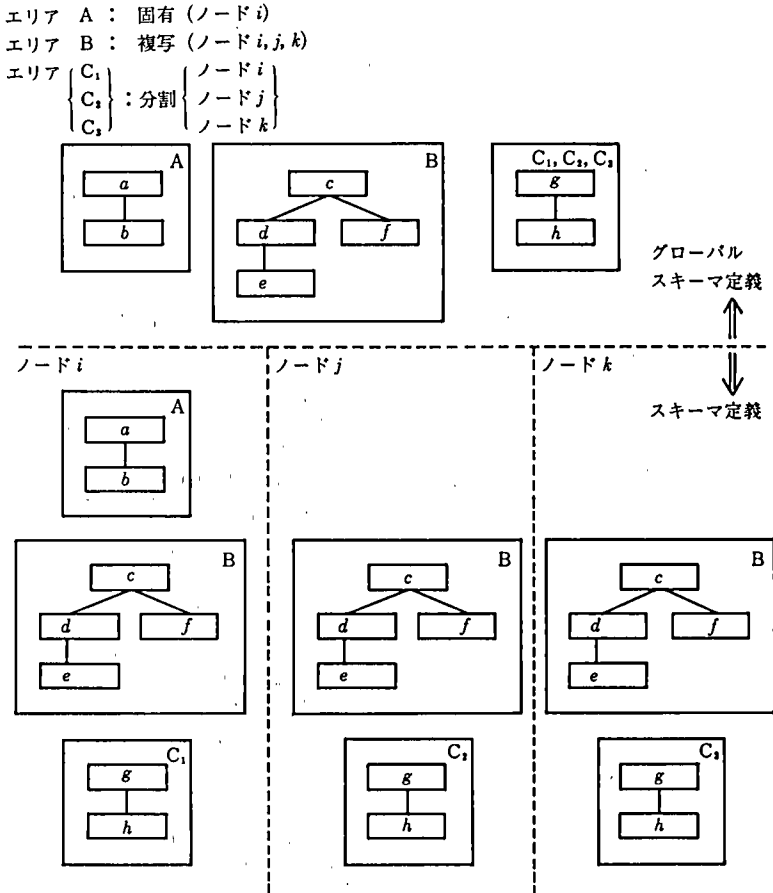


図11 分散データベースでのスキーマ定義の拡張
 Fig.11 Extension of distributed schema definition

3.2.1 NDD (Network Data Directry)

ASCOT 1100/RDA では、グローバル・スキーマ、グローバル・サブスキーマの概念を追加し、分散配置情報を NDD として展開する。NDD は AP から AIS 1100 に分散対象データのアクセス要求がなされたとき、アクセス・ノード決定のために参照する。

NDD に展開されている分散配置情報は表 1 および図 12 の通りである。

3.2.2 CRT (CuRency control Table)

ASCOT 1100/RDA は、分散データベース・システムでのカレンシ制御として CRT と分散 DBK (DataBase Key) を追加する。

CRT は各ラン・ユニットごとに管理され、DMS 1100 のカレンシ制御と同様に DB アクセスごとにラン・ユニット、レコード、エリア、セットのカレンシを更新する。

カレンシ情報としての DMS 1100 の DBK は、カレンシによってアクセス・ノードを決定する場合に備え分散 DBK として拡張される (図 13)。

データベースのアクセス・ノード選択方式は表 2 に示す 5 種類からなる。

表 1 NDD 分散配置情報
Table 1 Information of NDD

項目	登録情報	概略
ノード	ノード名、ノード ID ホスト ID サーバ TRXCD	分散システムに参加するノードのユニーク ID、名前 ノードの存在するホスト ID FCSS、テーブル用 TRX (トランザクション) コード
グローバル・スキーマ	G スキーマ名 ノード ID スキーマ名 サーバ TRXCD	分散システム下でのスキーマ名 DB が配置されるノード ID 各ノードでのスキーマ名 該スキーマのサーバ TRX コード
スキーマ	DMS1100スキーマ定義項目	DMS1100スキーマ定義
エリア	DMS1100スキーマ定義項目 ファイル・タイプ 配置ノード ID エリア ID	DMS1100スキーマ定義 ファイル・タイプ (分割, 複写, 固有) エリアの配置ノード AIS1100エリア ID
レコード	DMS1100スキーマ定義項目 レコード ID	DMS1100スキーマ定義 AIS1100レコード ID
セット	DMS1100スキーマ定義項目 セット ID	DMS1100スキーマ定義 AIS1100セット ID
グローバル・サブスキーマ	G サブスキーマ名 ノード ID サブスキーマ名	分散システム下でのサブスキーマ名 DB が配置されるノード ID 各ノードのサブスキーマ名
サブスキーマ	DMS1100サブスキーマ定義項目	DMS1100サブスキーマ定義
FCSS ファイル	FCSS定義項目 ファイル・タイプ ノード ID レコード番号範囲	FCSS定義 ファイル・タイプ (分割, 複写, 固有) ファイルの配置されるノード ID 各ノードのレコード番号範囲
システム・ テーブル	システム・テーブル定義項目 ファイル・タイプ ノード ID レコード番号範囲	システム・テーブル定義 ファイル・タイプ (分割, 複写, 固有) テーブルの配置されるノード ID 各ノードのレコード番号範囲

表 2 データベース・アクセス選択方式

Table 2 Access node selection

方 式	DMS1100コマンド
RID/AIDによるNDD参照	STORE record FETCH record
RIDによるNDD参照 オーナー・レコードCRT参照	FETCH via record STORE via record
SID/RID/AIDによるCRT参照	FETCH NEXT (within set) (within area)
R. U. (Run Unit) によるCRT参照	GET
分散DBKによるNDD参照	MODIFY record INSERT record (into set) REMOVE record (from set) DELETE record FETCH record

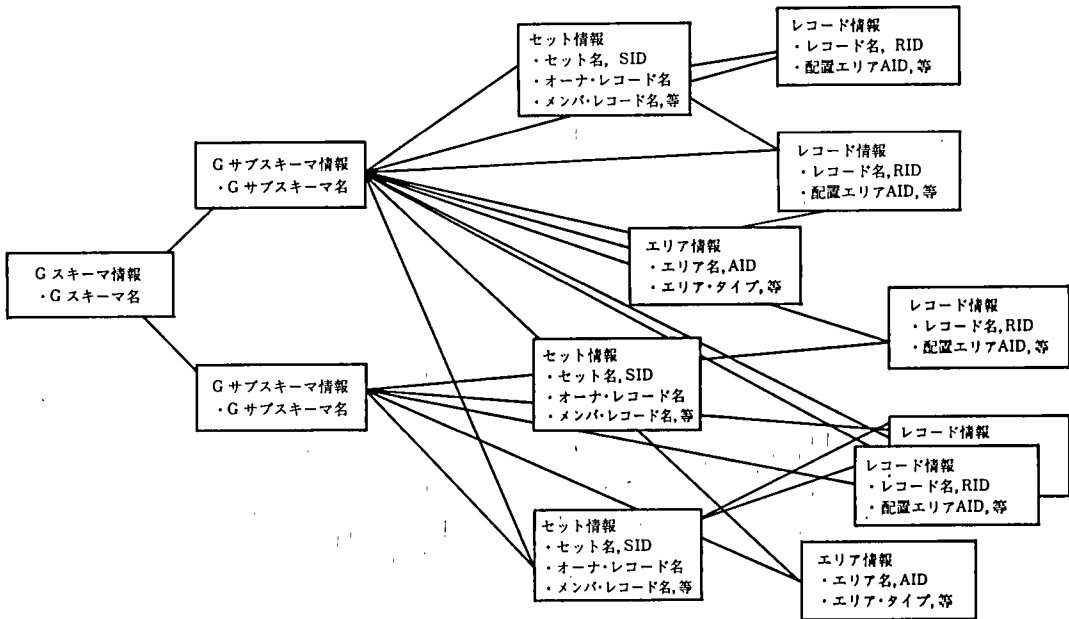


図 12 NDD のデータ管理

Fig. 12 Data management of NDD

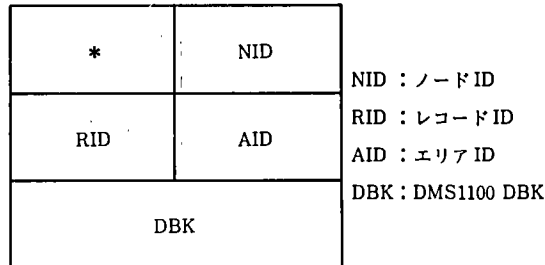


図 13 分散 DBK

Fig. 13 Distributed DBK

FCSS ファイル、システム・テーブルでのアクセス・ノードは、ファイル番号+レコード番号、テーブル番号+レコード番号から NDD を参照することにより選択する。例として、ある分割型、複写型データベースの参照、更新を示す (図 14, 15, 16)。

4. 分散アクセス制御

業務処理を実行する時に必要とするデータが自ノードになく、他ノードに存在する時には、他ノードにデータのアクセスを依頼することになる。他ノードへのデータ・リクエストの概要は図 17 のようになる。

4.1 マスタ・プロセスとスレーブ・プロセス

業務処理、リカバリ等の観点から、一つの処理単位を実行する時に協調する複数ノードに存在するプロセスにマスタ・プロセスとスレーブ・プロセスの区分を設ける。処理単位は、オンライン・システムでは、一つのトランザクション処理 (AIS/IR のリカバリ単位のステップと同じ) である。

マスタ・プロセスは業務処理を実行するプロセスであり、分散データ・アクセスの

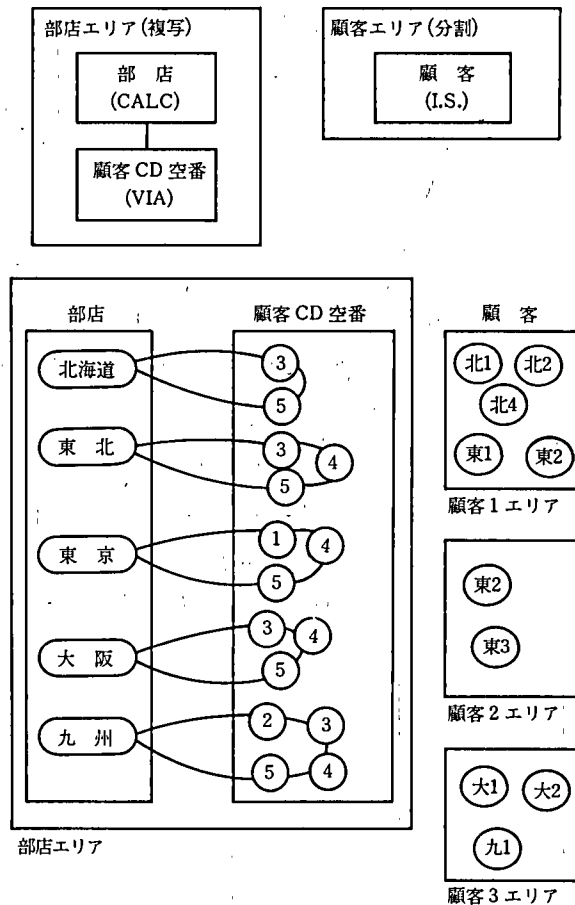


図 14 アクセス・ノード選択(例 1)

Fig. 14 Example of node selection 1

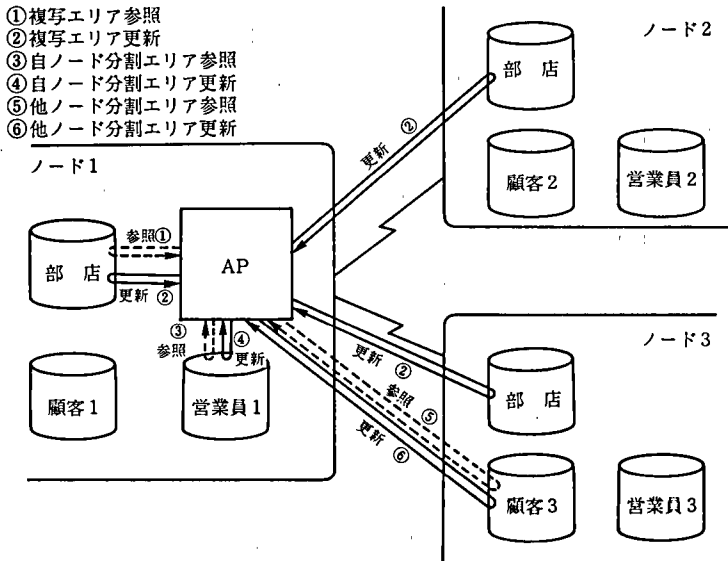


図 15 アクセス・ノード選択 (例 2)
 Fig. 15 Example of node selection 2

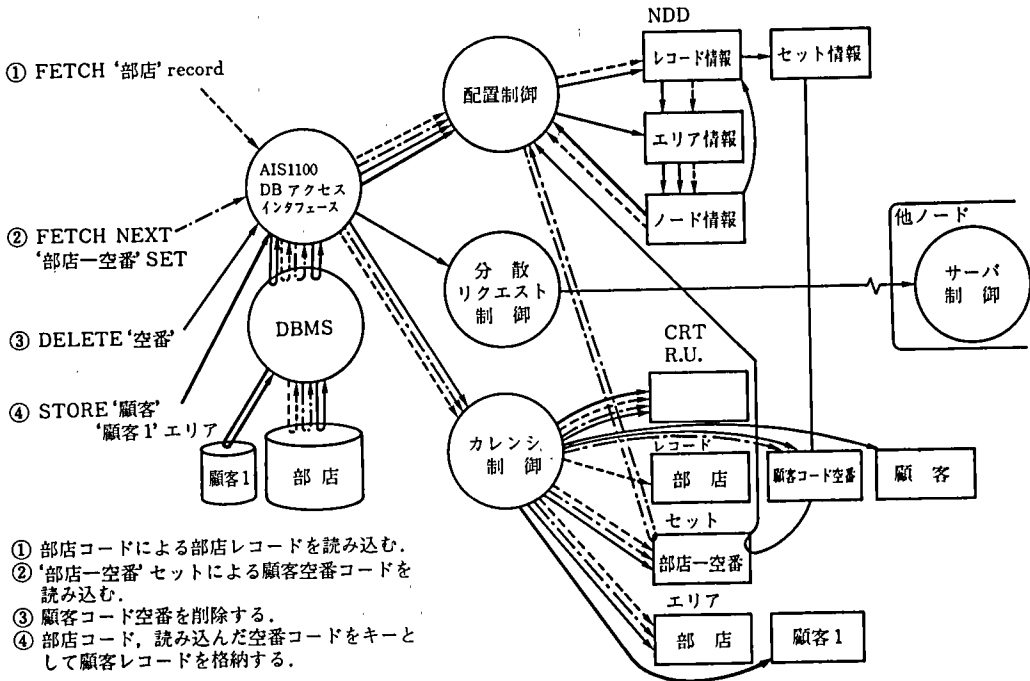


図 16 分散データ・アクセスの例
 Fig. 16 Example of remote data access

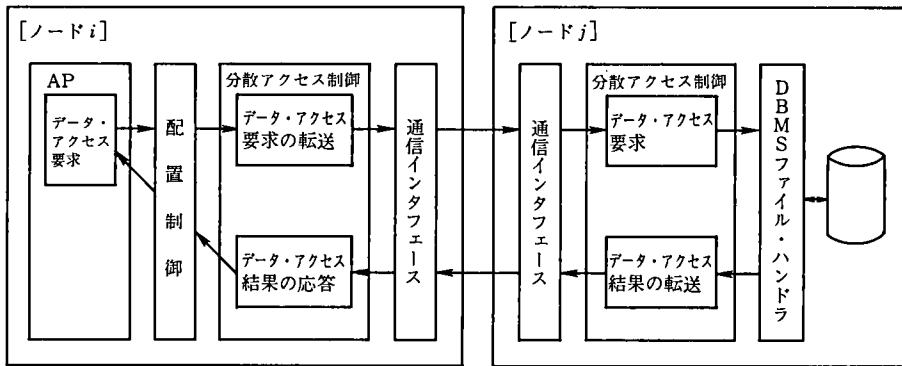


図 17 分散データ・アクセスの概要

Fig. 17 The outline of remote data access

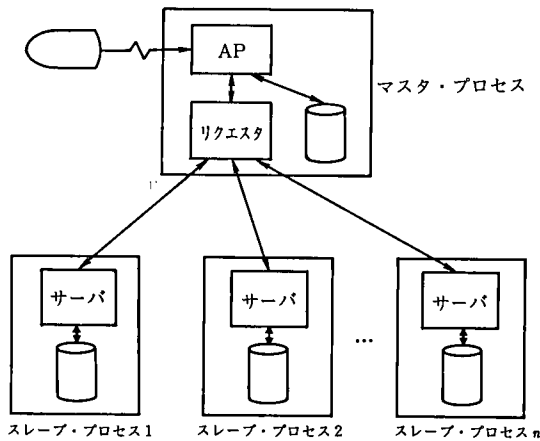


図 18 マスタ・プロセスとスレーブ・プロセス

Fig. 18 A master process and slave processes

観点から、リクエストがスレーブ・プロセスに対してデータ・アクセスや更新同期処理を依頼する制御を行う (図 18)。

マスタ・プロセスとスレーブ・プロセスの関係は 1 対 n となるが、スレーブ・プロセス間には関連を持たず、複数ノードとのデータ送受は必ずマスタ・プロセスがコントロールする。

スレーブ・プロセスではマスタ・プロセスの業務処理に連動してデータ・アクセスや更新同期処理を行う。スレーブ・プロセスとして ASCOT 1100/RDA には、サーバが存在する。サーバはマスタ・プロセスからの指示により開始・終了し、処理機能としては自ノード内のデータ (データベース、ファイル、テーブル) に対するアクセスとその処理結果の転送を行うデータ・サービス機能と更新同期制御機能がある。

4.2 分散データ・アクセス・プロトコル

マスタ・プロセス—スレーブ・プロセス間で分散データをアクセスするためのプ

ロトコルを分散データ・アクセス・プロトコルという(図19)。分散データ・アクセス・プロトコルは、OSIで応用層に位置付けられ標準化が進められているが、ASCOT 1100/RDAでは固有に定めている。

ASCOT 1100/RDAの分散データ・アクセス・プロトコルが定義するサービス(分散アクセス・マクロ)は表3の通りである。

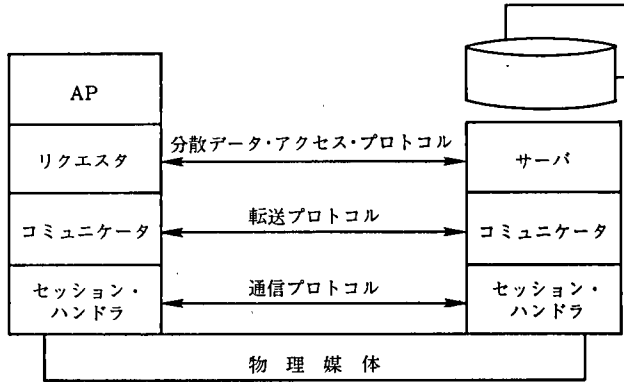


図19 分散データ・アクセス・プロトコル

Fig. 19 Remote data access protocol

表3 分散アクセス・マクロ
Table 3 Remote access macro

分散アクセス・マクロ	機能概要	OSIとの対応
VRMREQ	データ・アクセスと結果の応答を要求する。 データが存在するノードへのファースト・リクエスト時はトランザクションも開始させる。	[r-Associate+r-Open +C-BEGIN+] r-ExecuteDML
VRAPRP	コミット保証を指示する。	C-PREPARE
VRARDY	コミット可能を提示する。	C-READY
VRARFS	コミットを拒否する。	C-REFUSE
VRACMT	コミットを指示する。	C-COMMIT +r-Close
VRARBL	ロールバックを指示する。	C-ROLLBACK +r-Close+r-Release
VRACMP	トランザクションの終了を提示する。	r-Release
VRXRST	複数の凍結状態のトランザクションを再開始させる。	ΣC-RESTART

4.3 同時走行制御

複数のAPが走行可能な環境では、データの無矛盾性を保証するためにデータに対するロック・コントロールが必要であり、資源競合によるデッドロックが発生する可能性がある。

集中システムでは、DBMSやOS 1100/TIP等がデッドロックをチェックし、デッドロック発生時には、その解消のためにロールバック処理をする。分散システムにおいてもロック・コントロールは必須であり、デッドロックの検出方法を定める必要がある。

ASCOT 1100/RDA では、ネットワーク・ワイドのデッドロックの検出方法として時間監視方式を採用している。この方式には、次の利点がある。

- 1) システム負荷が少ない。
- 2) ノード内のデッドロックの検出に影響を与えない。
- 3) 必ずデッドロックが検出できる。

この方式は、自ノード内のデッドロック発生の有無の確認までは集中形 DBMS および OS 1100/TIP に依存し、分散アクセスに対してはマスタ・プロセスがロックウェイトの時間監視を設定し、設定時間内に応答がない場合にはネットワーク・ワイドなデッドロックが発生したものとみなし、スレーブ・プロセスをロールバックさせる。データベースへのリクエストの場合は、他ノードへの分散アクセス時にデッドロック・プライオリティも同時に指定する。

4.4 分散リクエスト制御

分散リクエスト制御はリクエストとサーバの間で行われる。リクエストは分散データをアクセスするためにサーバへ、コマンド、データ、カレンシ情報等を転送する。サーバはファースト・リクエスト時にコミュニケータからスケジュールされ、転送された情報に従い、データ・アクセスを行う。

複写型データベースの分散アクセスについては、データ操作命令の複合化、複数ノードでの並行処理を行い、とくに効率化をはかっている (図 20)。

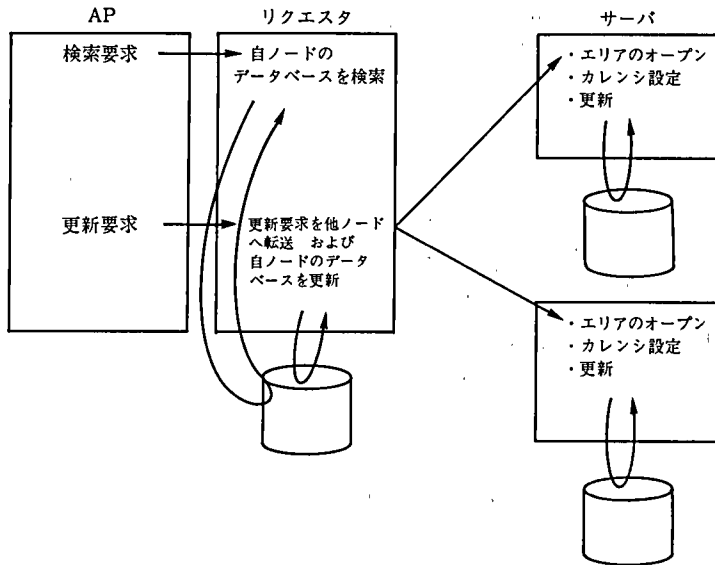


図 20 複写型データベースの分散リクエスト制御

Fig. 20 Remote request control of replicated database

4.5 セキュリティ

- 1) アクセス権のチェック……ASCOT 1100/RDA が対象とするデータに対して、アクセス権を NDD に登録可能とし、AP データ要求時にアクセス権のチェックを行う。
- 2) DMS 1100 セキュリティ機能との共存……DMS 1100 が提供するチェック・サ

ム/アクセス制御/ファイル・セキュリティ (EXEC 機能)等のセキュリティ機能との共存を図っている。

- 3) 暗号化……OSI等の国際標準の制定を待ち、当面は暗号化によるセキュリティ保証は行わないものとする。

5. 更新同期制御とリカバリ機能

更新同期制御・リカバリ機能は、更新要求が複数のノードにわたっている場合に、分散システムの各種障害に対してマスタ・プロセス、全スレーブ・プロセスを通しデータの一貫性を保証する。ASCOT 1100/RDA は、2 フェーズ・コミットメント方式を採用することにより上記の機能を実現している。

5.1 2 フェーズ・コミットメントによる更新同期制御

AIS 1100 の統合回復機能は図 21 に示すようにトランザクション処理のリカバリ単位をステップとし、遅延更新処理を採用することにより処理完了の要求 (同期点) で一括更新を行う (同期点処理)。また、ステップをステップ ID で特定し、ステップ情報 (コミット情報) を管理することにより、リカバリに備える。システム障害が発生した場合、コミット情報を基に IRU 1100 のショート・リカバリでシステムを回復する。

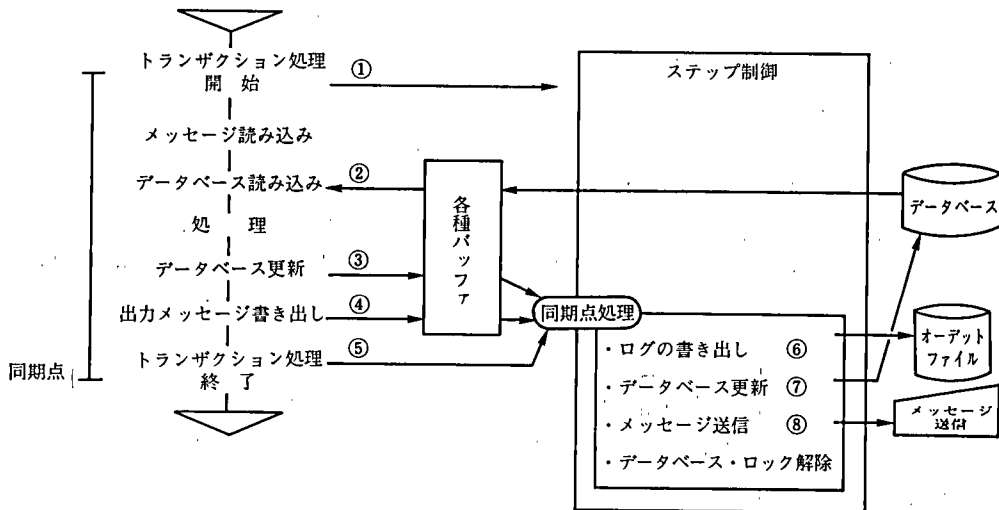


図 21 ステップ制御と同期点処理

Fig. 21. Step control and commitment

2 フェーズ・コミットメントでは、この同期点処理をコミット保証 (フェーズ 1) と実更新処理 (フェーズ 2) に分割し、フェーズ間でマスタ・プロセス主導で全スレーブ・プロセスとコミット保証の問い合わせを行う (PREPARE 命令)。スレーブ・プロセスからコミット保証の応答 (READY 応答) を受信した後、マスタ・プロセスがコミット宣言 (CIP 要求) を行い、全スレーブ・プロセスにコミット要求 (COMMIT 命令) を送信するとともに実更新する。スレーブ・プロセスはコミット要求を受信した後、実更新する (図 22)。マスタ・プロセスのコミット宣言がなされた時点でスレー

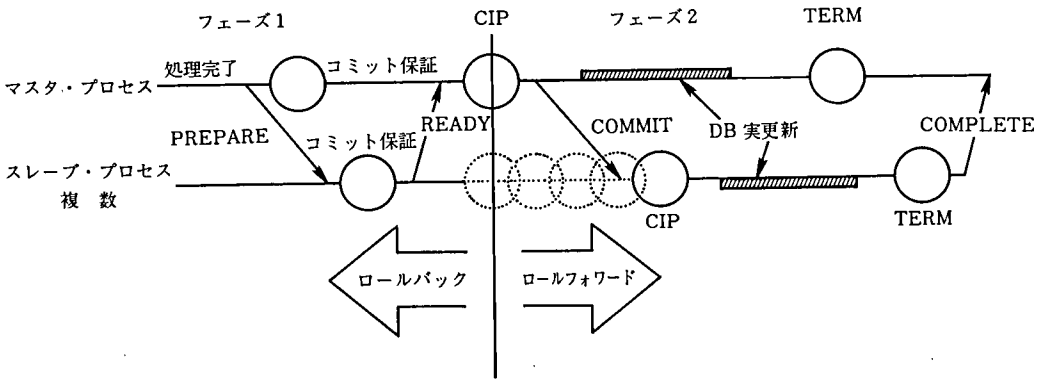


図22 2フェーズ・コミットメント
Fig. 22 Two phase commitment

ブ・プロセスも含め処理が完結したと判断し、分散システム内で障害が発生した場合でも、すべて処理を完結する。また、それ以外はすべてロールバック処理し分散システムでのデータの一貫性を保証する。

5.2 トランザクションの異常終了処理

APに異常（プログラム・エラー、I/Oエラー、コンティンジェンシ・エラー等）が発生した場合、AIS 1100のキャンセル命令、コンティンジェンシ・ルーチンから全スレーブ・プロセスに対してROLLBACK命令を出す。これによりマスタ・プロセス、全スレーブ・プロセスでロールバック処理が行われる（図23）。

サーバ（スレーブ・プロセス）に異常（I/Oエラー、コンティンジェンシ・エラー等）が発生した場合、サーバからAPにロールバック・ステータスが返される。APはAIS 1100のキャンセル命令を呼び出すことにより、APの異常終了と同様ROLL-

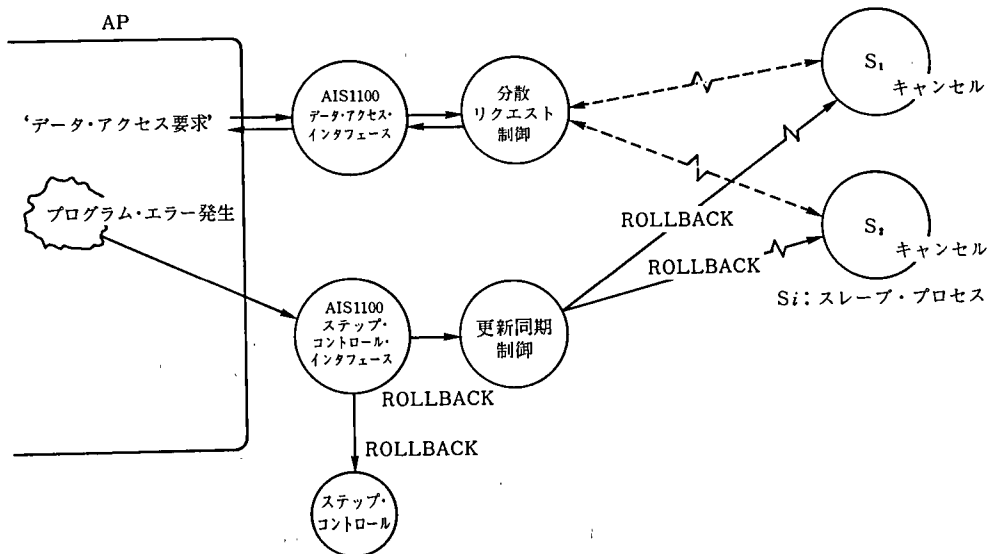


図23 マスタ・プロセス・エラーのロールバック処理
Fig. 23 ROLLBACK processing for the error of master process

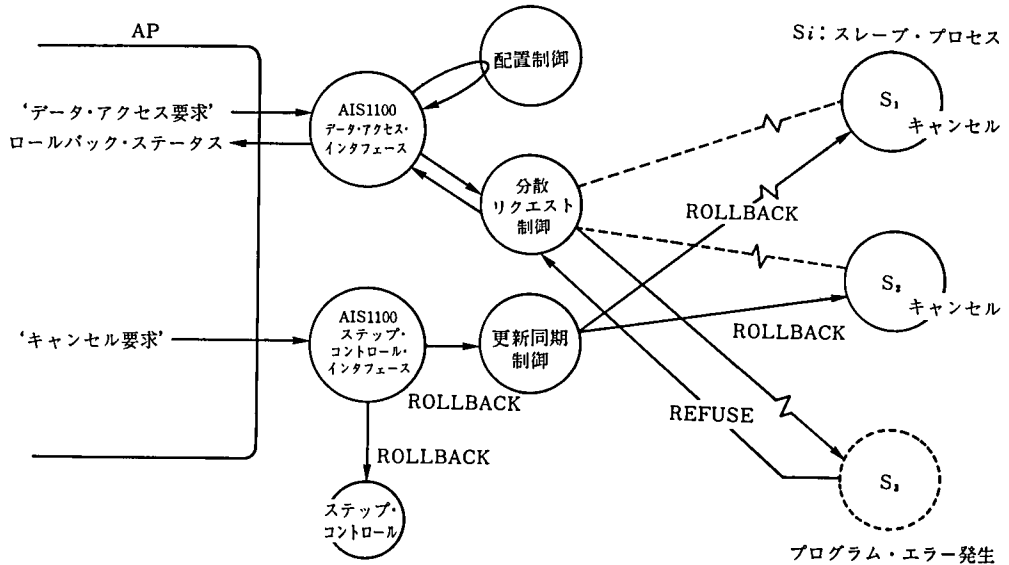


図24 スレーブ・プロセス・エラーのロールバック処理
Fig. 24 ROLLBACK processing for the error of slave process

BACK 命令を出しすべての処理をロールバックする (図24)。

また、ネットワーク・ワイドなデッドロックが発生した場合、マスタ・プロセス側同時走行制御のタイムアウト監視により、AP にデッドロック・ステータスが返される。AP は AIS 1100 のキャンセル命令を呼び出すことにより、AP の異常終了と同様 ROLLBACK 命令を呼び出しすべての処理をロールバックする (図25)。

5.3 システム障害のリカバリ

ASCOT 1100/RDA のコミュニケータによりマスタ・プロセスのコミット情報を入手し、マスタ、スレーブ・プロセスのステップ状態を一致させた後、IRU 1100 のショート・リカバリによりシステムを回復する。

5.3.1 スレーブ・プロセスのリカバリ

スレーブ・プロセスの存在しているノードが障害となったケースのリカバリ方式を図26により説明する。

- 1) マスタ・プロセスがスレーブ・プロセスから READY 応答を受信していない状態
 - ① マスタ・プロセス……すべてのスレーブ・プロセスに対して ROLLBACK 命令を出し、すべての処理をロールバックする。
 - ② スレーブ・プロセス (正常ノード)……マスタ・プロセスから ROLLBACK 命令を受信し処理をロールバックする。
 - ③ スレーブ・プロセス (障害ノード)……IRU 1100 のショート・リカバリにより処理をロールバックする。

①②は5.2節サーバ(スレーブ・プロセス)の異常終了と同様の処理である。
- 2) マスタ・プロセスが CIP を取得し、スレーブ・プロセスが CIP を取得していな

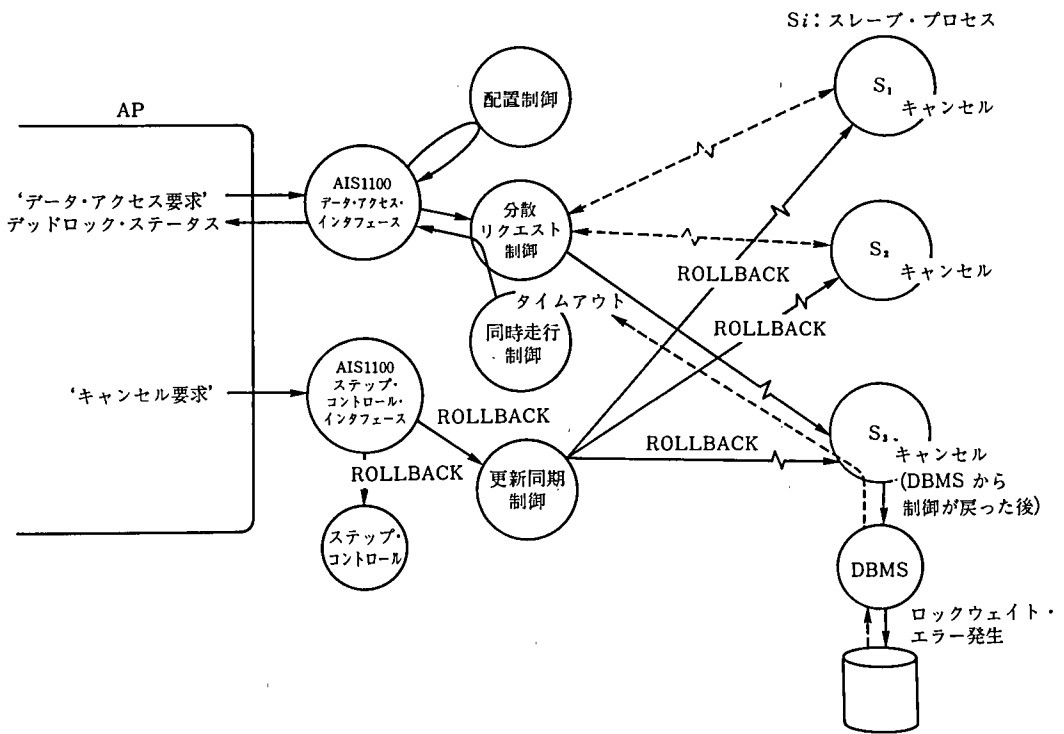


図 25 ネットワーク・ワイドなデッドロックに対するロールバック処理

Fig. 25 ROLLBACK processing for network wide dead lock

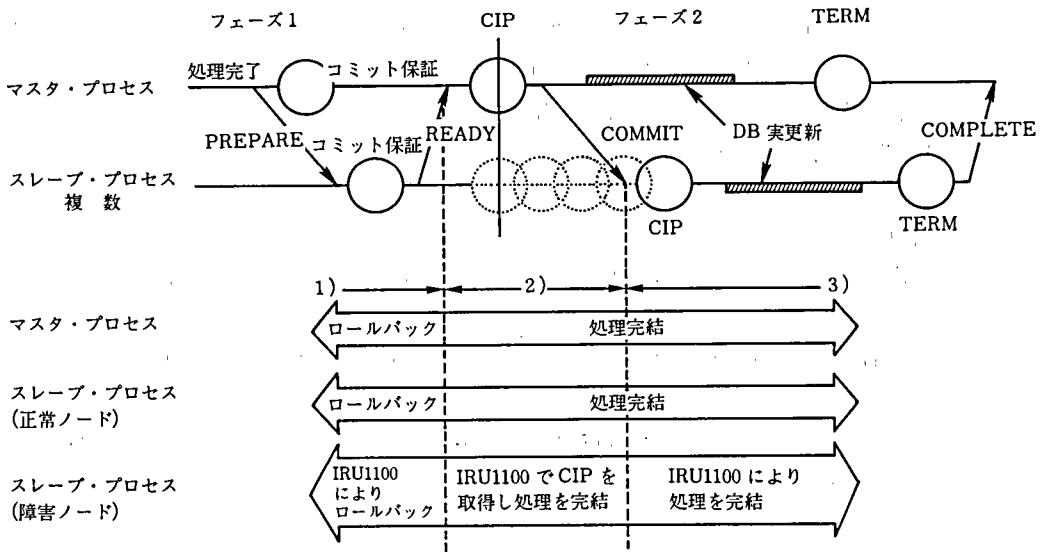


図 26 スレーブ・プロセスのリカバリ

Fig. 26 Recovery of slave process

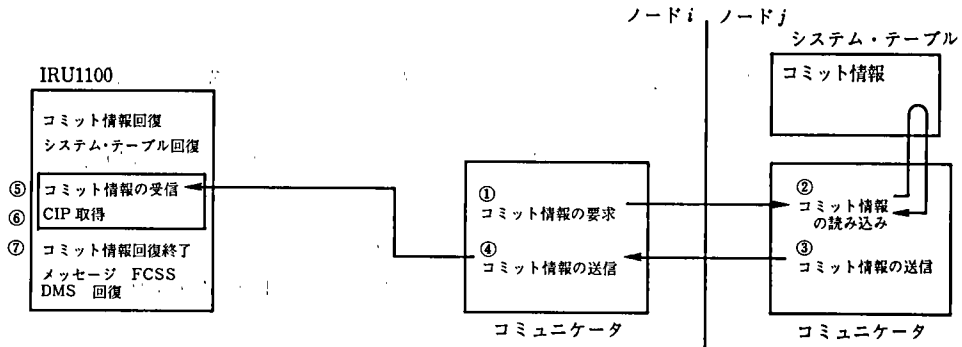
い状態

- ① マスタ・プロセス……障害スレーブ・プロセスに影響されことなく処理を完結する。
- ② スレーブ・プロセス (正常ノード)……障害スレーブ・プロセスに影響されことなく処理を完結する。
- ③ スレーブ・プロセス (障害ノード)……マスタ・プロセスで保存したコミット情報を取得し、IRU 1100 のショート・リカバリ実行時に CIP を取得し処理を完結させる。

3) スレーブ・プロセスが CIP を取得し実更新中の状態

- ① マスタ・プロセス……2)①と同様。
- ② スレーブ・プロセス (正常ノード)……2)②と同様。
- ③ スレーブ・プロセス (障害ノード)……IRU 1100 のショート・リカバリにより処理を完結させる。

障害となったノードのリカバリ制御を図示する (図 27)。



- ① 障害ノードのコミュニケータがコミット情報をマスタ・プロセスのコミュニケータに要求。
- ② マスタ・プロセスの保存したコミット情報をマスタ・プロセスのコミュニケータが読み込む。
- ③ マスタ・プロセスのコミュニケータが要求のあったノードのコミュニケータにコミット情報を送信する。
- ④ 障害ノードのコミュニケータは受信したコミット情報を IRU1100 に送信する。
- ⑤ IRU1100 はショート・リカバリ実行中でコミット情報を受信する。
- ⑥ IRU1100 は受信したコミット情報に基づき CIP を取得する。
- ⑦ IRU1100 はショート・リカバリを継続しリカバリを完結する。

図 27 障害ノード (スレーブ・プロセス) のリカバリ制御

Fig. 27 Recovery control of slave process

5.3.2 マスタ・プロセスのリカバリ

マスタ・プロセスの存在しているノードが障害となったケースのリカバリ方式を図 28 により説明する。

1) マスタ・プロセスが CIP を取得していない状態

- ① マスタ・プロセス……IRU 1100 のショート・リカバリにより処理をロールバックする。
- ② スレーブ・プロセス……マスタ・プロセスの障害がリカバリされるまで凍結状態とする。マスタ・プロセスのリカバリ後、受信したコミット情報に基づき再開し処理をロールバックする。

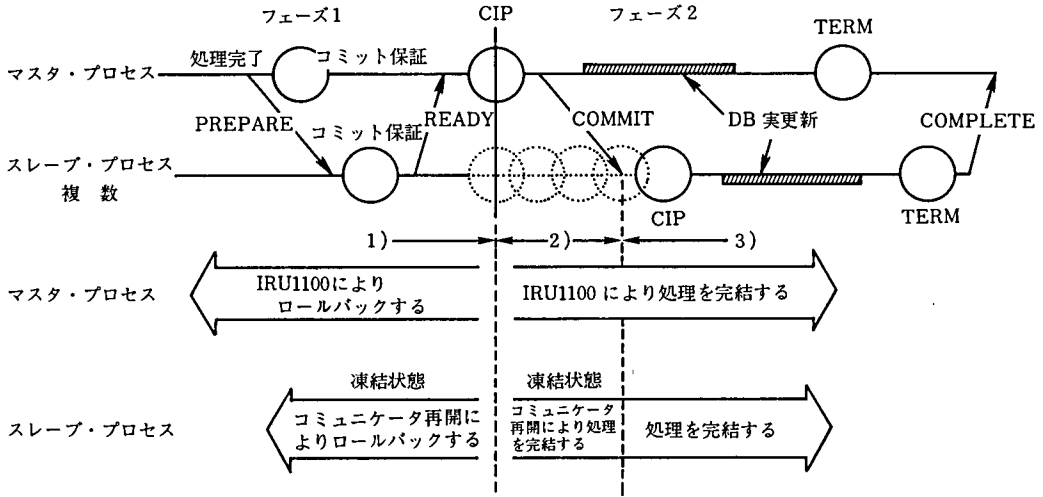


図28 マスタ・プロセスのリカバリ
Fig. 28 Recovery of master process

2) マスタ・プロセスが CIP を取得後、スレーブ・プロセスが COMMIT 命令を受信していない状態

- ① マスタ・プロセス……IRU 1100 のショート・リカバリにより処理をロールフォワードする。
- ② スレーブ・プロセス……マスタ・プロセスの障害がリカバリされるまで凍結状態とする。マスタ・プロセスのリカバリ後、受信したコミット情報に基づき再開し処理を完結する。

3) スレーブ・プロセスが COMMIT 命令を受信した状態

- ① マスタ・プロセス……2)①と同様
- ② スレーブ・プロセス……マスタ・プロセスに影響されことなく処理を完結する。

障害となったノードのリカバリと凍結状態のサーバの再開処理を図示する。障害となったノード (マスタ・プロセス) を IRU 1100 のショート・リカバリにより、リカバリした後、マスタ・プロセスのコミュニケータは、コミット情報をスレーブ・プロセスに送信する。これを受信したスレーブ・プロセスのコミュニケータは、凍結状態にあるサーバをコミット情報に合わせて起動する (図 29)。

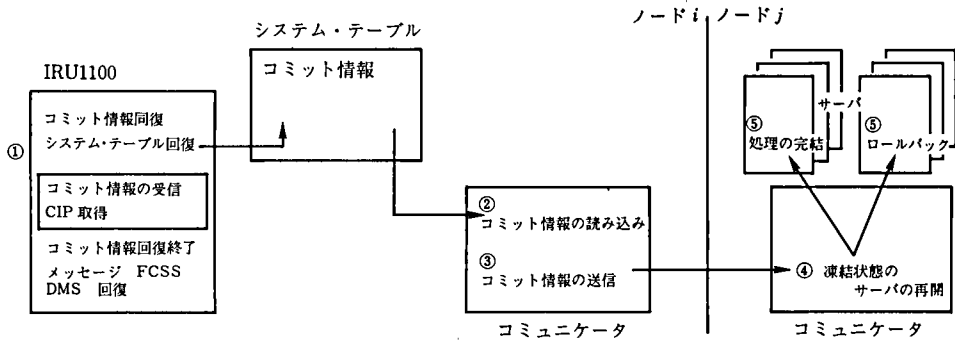
5.4 ファイル障害の局所化

データベースの障害は、そのデータベースの存在するローカルな DBMS (集中型 DBMS) によって管理される。集中型データベース・アクセスと同様に障害となったデータベースをアクセスする AP はロールバックされるが、そのデータベースを参照しない AP は何ら影響を受けない。また、ファイル障害からのリカバリはローカルに保持される更新後情報により集中型データベースと同様に回復できる。ただし、複写型データベースに関しては障害回復、運用が異なるので次節で述べる。

5.5 複写ファイルの維持・管理

AP からの複写ファイルに対する参照要求は、基本的にマスタ・プロセスの存在するノードで行う。AP からの複写型ファイルに対する一つの更新要求に対しては、分散システム内の全複写ファイルを自動的に更新する。複写ファイルの更新要求は固有型、分割型ファイルの他ノード・アクセスと同様の要求とみなし、前述の2フェーズ・コミットによりノード間の整合性を保障する。

複写ファイルの一つがファイル障害となった場合、全ノードにおける複写型ファイルの更新が不可となる。また、参照に関しても障害ファイルの存在するノードにマス



- ①IRU1100はマスタ・プロセスの保存したコミット情報(システム・テーブル)をリカバリする。
- ②マスタ・プロセスのコミュニケータはコミット情報を読み込む。
- ③マスタ・プロセスのコミュニケータはコミット情報をスレーブ・プロセスのコミュニケータに送信する。
- ④スレーブ・プロセスのコミュニケータはコミット情報に基づきサーバにROLLBACK, COMMIT 命令を出す。
- ⑤サーバはスレーブ・プロセスのコミュニケータから受信した命令に基づき処理を完結するかまたはロールバックする。

図 29 障害ノード (マスタ・プロセス) のリカバリと制御スレーブ・プロセスの起動

Fig. 29 Recovery of master process and restart server

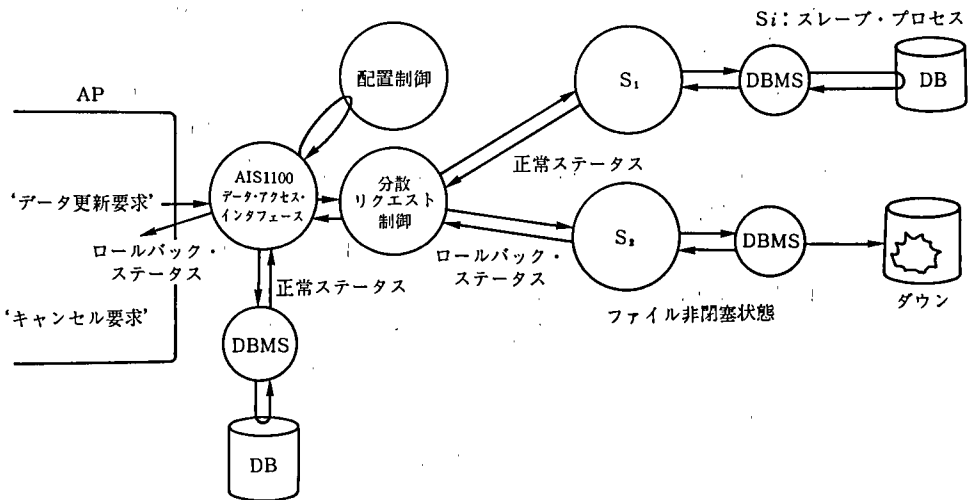


図 30 複写型ファイル障害時の更新

Fig. 30 Updation of replicated file at failure

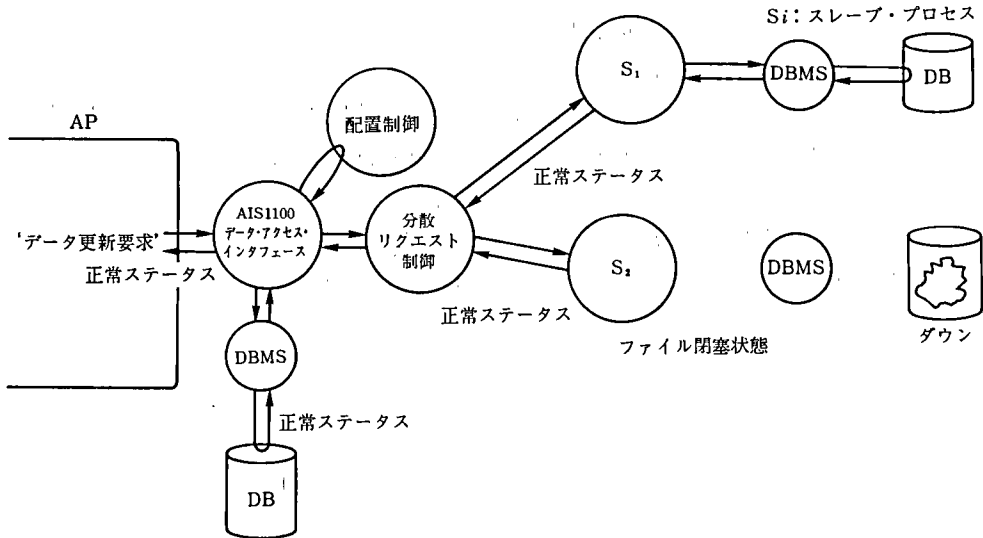


図31 複写型ファイル閉塞時の更新

Fig. 31 Update of replicated file while closing

タ・プロセスが存在する場合はアクセス不可となる（図30）。

この場合、障害ファイルをNDD上で閉塞状態に変更し、該当ファイルを分散システムから切り離すことにより障害ファイルをないものとみなし、再び更新が可能となる。また、障害ファイルの存在するノードのマスタ・プロセスは、切り離された複写ファイルの参照を他ノード上の当該複写ファイルの参照により満足する（図31）。

6. 運用管理

運用管理は分散システムでの構成管理と開始・終了処理、NDD操作およびそれに伴うオペレーション・インタフェース等の分散環境の運用管理を支援する。

6.1 分散システムの構成管理

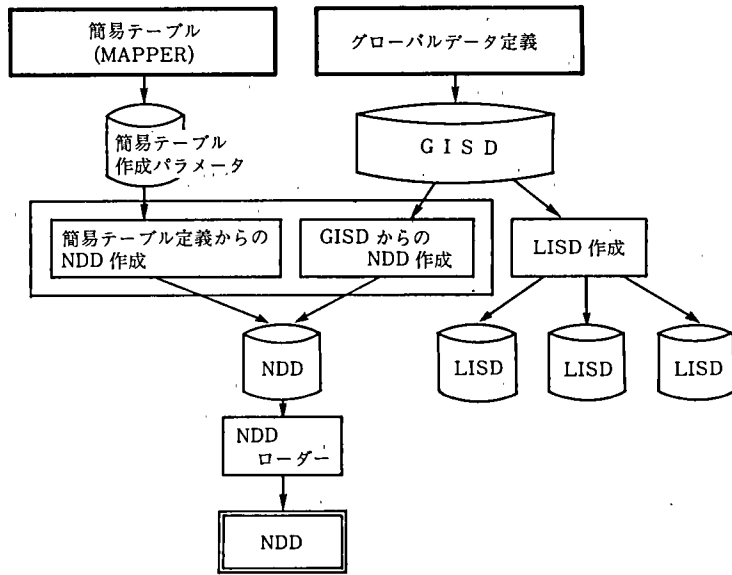
AIS 1100の提供しているISDを拡張し、全分散環境内のシステム構成情報（ネットワーク情報、分散データ情報）を一元管理する。この拡張したISDからAIS 1100のISDパラメタ、3.2節で述べたNDDの登録情報を生成する（図32）。

6.2 オペレータ・インタフェース

オペレータ・インタフェースとして、AIS 1100のオンライン・コンソール機能にASCOT 1100/RDA標準コマンドを追加しており、使用者は容易に分散システムに対して操作できる。ASCOT 1100/RDA標準コマンドは、分散システム内のどのオンライン・コンソールからでも入力でき、他ノードのシステム操作が可能となっている。

ASCOT 1100/RDA標準コマンドには、次の2種類がある。

- 1) システム操作……ASCOT 1100/RDAシステム終了、稼働状況表示、分散アクセス・ログ採取等
- 2) NDD操作……NDDデータ定義情報の参照、ノード・ステータスおよびファイル・ステータスの変更



GISD(Global ISD)：分散システム内での複数ホストの統合システム辞書
 LISD(Local ISD)：単一ホストでの AIS1100 統合システム辞書

図 32 分散構成定義

Fig. 32 Definition of distributed system

7. おわりに

分散データベース技術は、コンピュータ・システムの大規模化、多様化が進むにともない、その重要性は今後いっそう増すことが予想される。ASCOT 1100/RDA の機能は、分散データベース技術の進歩や標準化および市場ニーズの動向に対応して、成長し続ける必要がある。

表 4 は、分散データ・アクセスの要件に対する現在の ASCOT 1100/RDA の機能の充足度を表している。

表 4 ASCOT 1100/RDA の機能の充足度
 Table 4 The evaluation of ASCOT1100/RDA's function

分散データ・アクセスの要件	充足度
分散を意識しないアクセス(分散不可視性)	○
複数ノードへの同時アクセス	○
分割型ファイルのサポート	○
複写型ファイルのサポート	○
データ保全性の保証	○
動的な構成変更	×
個々のノードの独立運用	○
運用の遠隔地操作	○
異機種接続	×

今後の課題として、次に示す機能の追加があげられる。

- 1) 動的構成変更
- 2) リレーショナル・データベースの対応
- 3) 異機種接続

本プロダクト開発で、日本ユニシス(株)のソフトウェア・プロダクト本部ソフトウェア1部、ソフトウェア2部の方々に多大なる協力をいただいたことに深く感謝の意を表する。

- 参考文献
- [1] G. A. Champine, “分散コンピュータ・システム”, 企画センター, 1981.
 - [2] 塚本・八田編, 大特集“分散処理技術”, 情報処理, Vol.28, No.4, p.369, 1987.
 - [3] B. W. Leong-Hong・B. K. Plagman, “データディクショナリ/ディレクトリシステム”, オーム社, 1985.
 - [4] 上林・Kim・酒井編, “最近のデータベースシステムとその応用”, 第6章・分散データベース・システム, bit 別冊, 1983.
 - [5] 森・吉田・鈴木・伊藤, “分散形データベース管理システムの実用化”, 通研実報, pp.2841~2853, 1984.
 - [6] 河津・紫崎・南・大沼, “DCNA のデータベース・アクセス・プロトコル”, 通研実報, p.799, 1981.
 - [7] J. N. Gray 他編, “フォールト・トレラント・システム”, 第9章・グローバル更新プロトコル, マグロウヒルブック, 1986.
 - [8] W. Rauch-Hindin, “分散データベース管理システム, 分散照会などの問題を抱えながらも製品化が進む”, 日経エレクトロニクス, pp.281~292, 8月24日, 1987.

執筆者紹介 福島 康夫 (Yasuo Fukushima)

1983年早稲田大学理工学部工業経営科卒業。同年日本ユニシス(株)入社。統合オンライン・ソフトウェア AIS 1100の開発, サービスに従事。現在ソフトウェア企画開発部企画室に所属。



米津 政紀 (Masanori Yonetsu)

1983年早稲田大学理工学部数学科卒業。同年日本ユニシス(株)入社。システム効率のコンサルティング, ホット・スタンバイ・システムの構築支援を経て, プラットホーム・プロダクトの企画, 開発に従事。現在ソフトウェア企画開発部企画室に所属。情報処理学会会員。



ESD におけるエネルギー変換とその妨害作用

Energy Dissipation in Electrostatic Discharge, and its Interference Effects

本 田 昌 實

要 約 空気中における金属物体間での静電スパーク放電に関し、放電前後のエネルギー（電荷量）の変化と、このとき金属物体の近傍に生ずるインパルス電磁界の性質について実験を行った。

その結果、放電により電荷損失が生じること、そして ESD 耐性試験時に考慮すべき妨害作用の基本的な特性のいくつかが明らかとなったので報告する。

Abstract This paper describes some results of an experiment on electrostatic spark discharge between metallic bodies before and after discharge takes place.

The experiment has clarified some of the properties of impulsive electromagnetic fields produced in the vicinity of the body and some of the basic points of EMI effects to be considered for ESD testing.

1. はじめに

帯電物体に集中して蓄えられていた静電エネルギー ($W_0 = CV^2/2$) は、空気中でのスパーク放電によってその姿を変え、熱、光、イオン化そして電磁(界)エネルギー等に変換される^[1]。

とくに静電気放電（以下 ESD と略す）の熱的な作用に関しては可燃性気体、液体、粉体等を扱っている産業において爆発事故等の致命的な災障害の原因になるため、従来から多くの研究が成されてきた。文献^[2]によると、静電エネルギー W_0 が $100\mu\text{J} \sim 2,000\mu\text{J}$ （容量 C が $28\text{PF} \sim 194\text{PF}$ ）の範囲で Constant-Volume Calorimeter を用いた測定ではスパーク放電に際しその 95% 以上が熱に変換されると報告されており、これらの研究を基に水素ガス、ガソリン等の最小着火エネルギーが算出され、危険防止のための静電気管理が行われている。

しかしながら、エレクトロニクスの分野における ESD の研究は静電気の発生と帯電のプロセスまでは石油産業と共通の認識が行われているにもかかわらず、放電に際しては静電(気)エネルギーの 100% が電磁エネルギーに変換されるものとして議論が進められているところに重大な問題（前提条件の違い）がある。しかもこの電磁エネルギーの伝達と消費の対象は、LCR 集中定数（回路）解析に見られるように一次元の長さだけを持った導体（物質）を想定しており、電子機器や半導体素子が現実に遭遇する ESD の世界と大きく異なる。

LCR 集中定数を用いた回路の特性表示は、以下の基本的な前提条件がその背後にあるということを忘れてはならない^[4]。

- 1) すべての電界 E はキャパシタの内部に閉じこめられている。
- 2) すべての磁界 H は Coil の内部に閉じこめられている。

3) その回路の大きさ (寸法) は考察対象の波長に比較して十分小さい。

2. 現実の ESD の世界

半導体素子やこれを搭載した回路基板 (以下 PCA と略す) に、人がまったく手を触れていないのに VLSI が ESD 損傷を被っていることがしばしば経験される。

また CRT 端末機の操作中にオペレータが椅子 (部分的に金属性) から立ち上がったところ突然にディスプレイが消えてしまうトラブルも発生している。

前者の説明には Charged Device Model (以下 CDM と略す) が用いられ、人体の帯電を前提とした Human Body Model (以下 HBM) は不適切である。そして後者のトラブルに至った原因の説明に際しても、HBM の適用は現実の状況との間にいくつかの矛盾があるため非常な無理があると言わざるを得ない。

それらは、

1) CRT 端末機のキーボードや収容キャビネットは絶縁抵抗の高いプラスチックでできており、人体・椅子からの放電電荷を直接受け取ることができない

2) 床はカーペットまたは P タイルでできているため放電電荷 (電流) の通り道である “Ground Return” が存在しない^[5]

3) 人体および椅子と電子機器の間には空間がある
等であり、帯電した人体の指先から放電に際して対象の装置に直接的に電荷の移動が行われるものとした HBM と本質的に異なる。

半導体素子や PCA、電子機器における ESD 耐性試験・評価の際の Immunity Level は HBM にせよ CDM にせよ、また多くの他のモデルがそうであるように一般には電圧 (V) またはエネルギーの大小をもって表示されている^[6]。しかしながら、現実の ESD の世界ではそのエレクトロニクスが工場 (または設計時) で保証された Immunity Level よりはるかに低いエネルギーで素子の破壊や電磁妨害 (以下 EMI と略す) 問題が発生している^{[7],[8]}。

このエネルギーの大小による ESD Immunity Level の表示が、なぜ現実の ESD の世界で経験される状況と大きく食い違うかを明らかにしなければならない。その手がかりのいくつかは、次のごときものであろう。

1) 静電スパーク放電は宇宙システムを除き、空気中でのみ発生し、その放電点は作用を受ける (与える) 物質の表面上にある。

2) 静電スパーク放電によって静電エネルギーは消費され変化する。

3) 導体上のエネルギー伝送ばかりではなく、空間もまたエネルギーの伝達媒質となる。

この中でとくに 3) に関しては、筆者らが従来から問題視してきた “Indirect ESD”^[7] による電子機器に対する EMI メカニズムを解き明かす上で、根幹的な意味合いを持つものと考えられる。

3. 実 験

ESD におけるエネルギー変換と、その時生じる電磁界の性質を調べるため主に空間を意識した実験を行った。

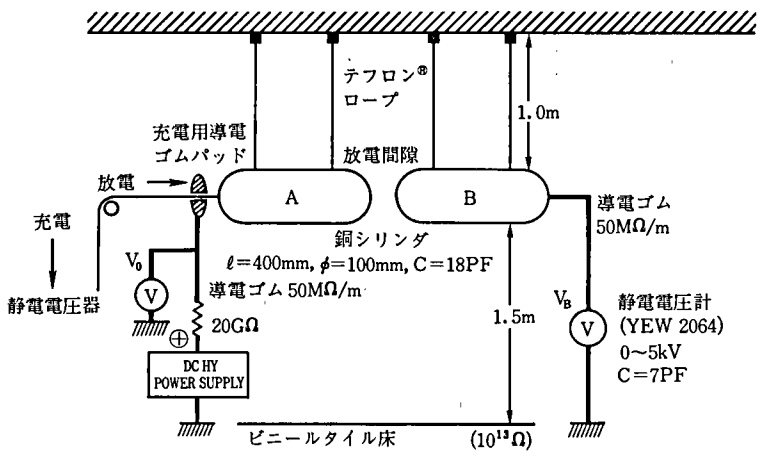


図 1 エネルギー変化検証実験構成

Fig.1 A setup experiment to verify energy dissipation by ESD

3.1 スパーク放電におけるエネルギー消費

帯電物体の有する静電エネルギー $W_0 = CV^2/2$ または $QV/2$ がスパーク放電によって、非帯電物体へどのくらい転送されるのかを調べるため図1のごとき構成にて実験を行った。

銅製シリンダ(長さ $l=400$ mm, 直径 $\phi=100$ mm, 静電容量 $C=18$ PF), A, B の 2本を天井から 1 m の空中に水平に吊す。シリンダ A を直流高電圧電源(電流制限抵抗 20 G Ω 付)に接続されている導電ゴムパッド (50 M Ω /m) に接触させ電源電圧 V_0 に充電したのち, 1 mm/sec 以下の非常にゆっくりとした速度でシリンダ B に接近させスパーク放電だけを発生させる。放電が発生したらただちに A を停止し, B に接続されている静電電圧計 (YEW 2064, $C_m=7$ PF) の目盛り V_B を読む。同一 V_0 においてこれを 10 回試行しデータを得た。シリンダ A, B の電荷漏洩, コロナ損失を最小にするため A, B の支持は天井から吊したテフロン®ロープ (絶縁抵抗 10^{15} Ω 以上) だけで行った。またシリンダ A, B への高電圧印加/接地および静電電圧計への配線は, すべて導電ゴム (50 M Ω /m) を用い安定した直流高電圧の供給とスパーク放電に際しての放射アンテナ作用の防止を計った。

実験の結果, 図2に示すグラフが得られた。シリンダ A から B へスパーク放電のみで電荷を転送した場合, B の電圧 (V_B) は理論値 ($V_{B0}=0.42V_0$) に達せず高電圧になるに従い開離の傾向を示す。またシリンダ A と B をスパーク放電だけではなく, 完全に接触させたとき (図2のグラフ中・印) も V_B の値は V_{B0} に一致せず電源電圧 V_0 の増大に伴い開離の傾向を強める。

図3にシリンダ B がスパーク放電によって実際に受け取った電気量 $Q_B(V_B \times (C_B + C_m))$ と本来受け取るべき電気量 $Q_{B0}(K \times V_0 \times (C_B + C_m))$ との比, すなわち転送効率 η_0 を示す。

$$\eta_0 = \frac{Q_B}{Q_{B0}} \times 100\% = \frac{V_B \times (C_B + C_m)}{K \times V_0 \times (C_B + C_m)} \times 100\%$$

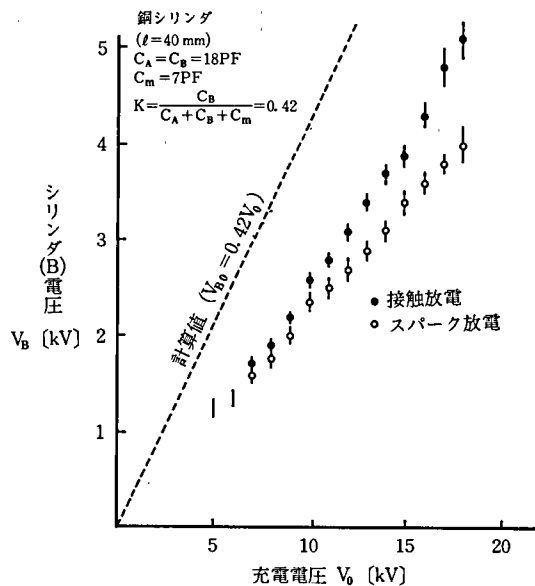


図 2 充電電圧とシリンダ(B)電圧の関係
Fig.2 Cylinder voltage vs charge voltage

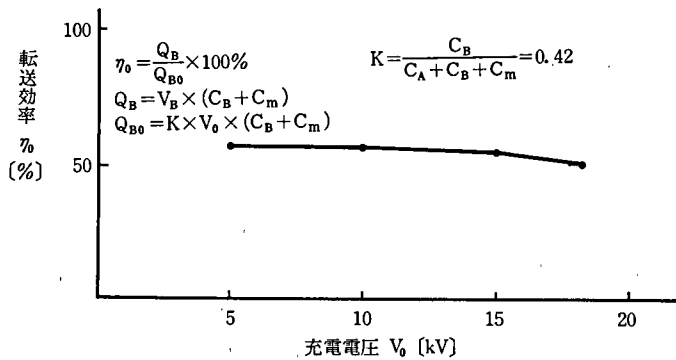


図 3 スパーク放電発生時の電圧と電荷転送効率の関係
Fig.3 Transmitted charge vs spark discharge voltage

$$K = \frac{C_B}{C_A + C_B + C_m} = 0.42$$

このグラフから V_0 が 5 kV ~ 12 kV の範囲でシリンダ B は約 57 % しか電荷を受け取っていないが、 V_0 が 15 kV 以上では本来 (計算値に対して) 受け取るべき電気量の約 53 % しかない。すなわち、高電圧 (5 kV 以上) のスパーク放電においては電荷損失が 43 ~ 47 % も発生していることがわかる。

このシリンダ A, B を用いた実験では電荷を蓄える静電容量 C_A, C_B が 18 PF しかなく、この値は測定系 (静電電圧計 YEW 2064) の内部容量 $C_m = 7\text{PF}$ と比較しうる量で

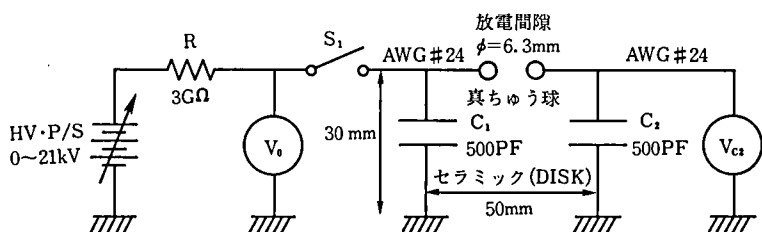


図 4 集中定数回路における電荷転送の実験回路

Fig. 4 Set up for experiment of charge transfer between capacities as lumped constants

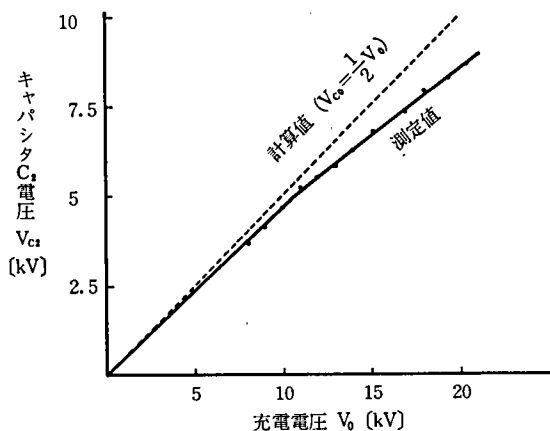


図 5 充電電圧キャパシタ (C₂) 電圧の関係

Fig. 5 Capacitor voltage vs charge voltage

あるため、改めて図 4 の回路にて示す通り静電容量 500 PF のキャパシタ (セラミック円板型) C_1 , C_2 を使い、先と同様の実験を行った。この回路は机上にあり、全体を極力小型化し C_1 , C_2 のリード部を除くすべての電線はビニール被覆電線 AWG # 24 を用いた。

キャパシタ C_1 は S_1 を閉じるにより電源電圧 V_0 に充電される。 S_1 を開き C_1 の先端に付いている真ちゅう球 (直径 $\phi = 6.3$ mm) をテフロン棒を用い、 C_2 の先端に付いている同様の真ちゅう球へゆっくりと接近させ、1 回のスパーク放電を発生させる。そしてこのとき、ただちに C_2 に接続されている静電電圧計 ($C_m = 7$ PF) の値 V_{c2} を読む。

この結果図 5 のグラフに示す通り、 C_2 の電圧 V_{c2} は電源電圧 V_0 の上昇に従い計算値 ($V_{c2} = V_0/2$) からわずかに下回る傾向が出てくるが、この値は V_0 が 20 kV においても 5 % 未満であった。このような実験条件では、キャパシタ間でのスパーク放電による電荷の転送は先の実験 (シリンダ A, B) と比較してあまり大きな損失を伴わずに行われるということがわかった。

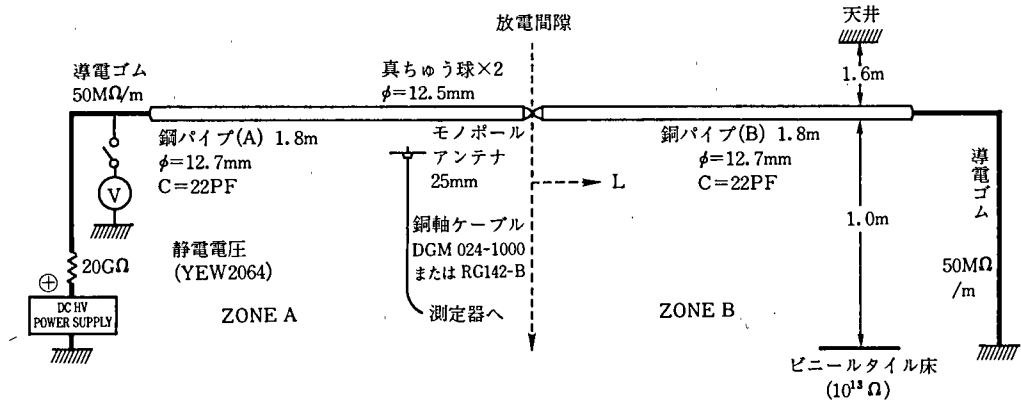


図 6 インパルス電磁界測定のための実験装置

Fig. 6 Setup for experiment to measure impulsive electromagnetic fields

3.2 スパーク放電点からの距離の違いによる電磁界の変化

銅パイプを用い、図6のごとき放電治具を構成し放電点（スパーク・ギャップ部）から空間を隔てたある場所における電磁界の様子を広帯域オシロスコープ（時間領域）、およびスペクトラム・アナライザ（周波数領域）を用いて観測した。

2本の銅パイプ（ $\ell=1.8$, $\phi=12.7$ mm, $C=22$ PF）A, Bを必要な間隔を確保してテフロン・ロープおよびアクリルの支持材を用い、床上1 mの所に水平に固定した。銅パイプAの荷電は直流・高電圧電源のプラス出力を電流制限抵抗20 G Ω を介して導電ゴム（50 M Ω /m）により、たえず行っている。銅パイプA, Bの先端には真ちゅう製球（ $\phi=12.5$ mm）が取り付けられており、この先端部・球電極にてスパーク放電を間欠的（例： $V_0=+1.5$ kVのとき約3回/秒）に発生させる。銅パイプBのもう一方の端は、電荷の緩慢な漏洩を行うため導電ゴム（50 M Ω /m）を介して大地へ接地している。

このように高電圧と接地電線の引き回しを、すべて高抵抗の導電ゴムにより行ったためスパーク放電に際しての物体からの電磁波の放射は2本の銅パイプA, Bからのみとすることができた。

銅パイプA, Bの近傍の空間に現われる電磁界の検出にあたっては、種々の理由により長さ25 mmのモノポール・アンテナ（50 mm \times 50 mmのグランドプレーン付）を用い、これに原則としてマイクロ波用（DC \sim 12 GHz）の長さ1 mのフレキシブル同軸ケーブル（DGM 024-1000, $Z_0=50$ Ω ）をSMAコネクタで接続した。

電磁界の時間領域の観測には、有効帯域幅1 GHzのリアルタイム・オシロスコープ（TEKTRONIX 7104, 7A29, 7B15, $tr=0.4$ ns）を用いた。強力な電磁界による測定上の障害を除去するため、測定器本体を三重の銅製ネットでシールドし接地した。

3.2.1 時間領域での測定結果

銅パイプA, B近傍の空間に図7のごとき座標系（測定点A₁~A₆, B₁~B₆）を採り、この各点において先のモノポール・アンテナを高さ1 mの所に垂直に配置し、スパーク放電点からの距離の違いによる電界の変化を調べた。この一連の実験においては銅

放電間隙 (L=0, d=0, h=1.0m)

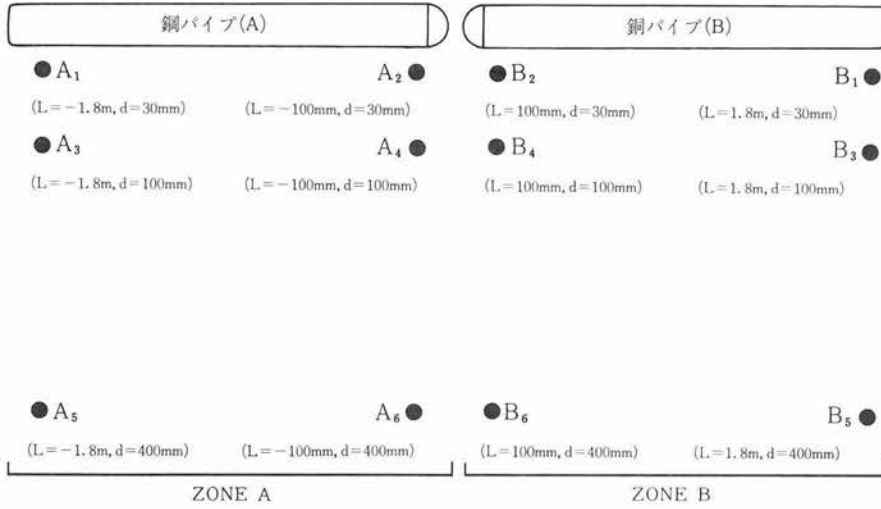


図 7 測定見取図

Fig. 7 Arrangement plan of measuring points

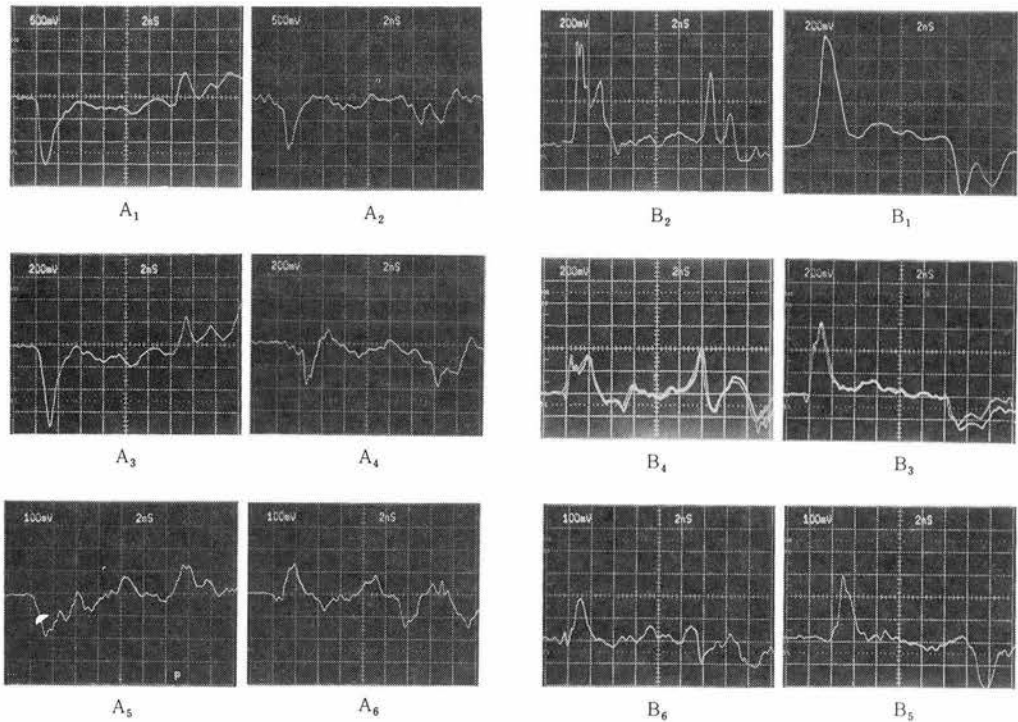


図 8 異なった距離によるインパルス電磁界の変化

Fig. 8 Changes of impulsive electric field with distance (V 軸: 電圧, H 軸: 時間)

パイプ A に絶えず $V_0 = +1.5 \text{ kV}$ を印加した。

その結果図 8 に示すごとく、 $A_1 \sim A_6$ (ZONE A), $B_1 \sim B_6$ (ZONE B) のいずれの各点においても極性を有するパルス状の電界が認められる。その極性は放電の瞬間 ZONE A の空間ではマイナス方向への電界の変化が発生し、ZONE B ではプラス方向の変化が空間に現われる。

この明確にとらえられる極性の反転は非常に単純であり、それは放電点(スパーク・ギャップ部)の手前(すなわち帯電物体側)の空間か、または放電点より後(放電電荷を受け取る側)の空間かによってのみ決定される。

放電を引起こした物体の近辺の空間に現われる電界の極性は、帯電物体側であればその帯電電荷の極性と反対になり、放電電荷を受け取る側であれば帯電電荷と同じ極性が生じる。

3.2.2 周波数領域での測定結果

銅パイプ B の先端(球状電極部)から $L = 10 \text{ mm}$ の距離において、このパイプの直上 $d = 2 \text{ mm}$ のスペクトラム分布と、 $d = 40 \text{ mm}$ の空間におけるスペクトラム分布を調べた。このとき、銅パイプは $V_0 = +0.5 \text{ kV}$ を印加しており、モノポール・アンテナの受信高さはいずれの点も $h = 1.0 \text{ m}$ とした。

その結果、図 9 に示すごとく銅パイプ B の直上と空間を距離 40 mm 隔てた点では、スペクトラム分布、受信電力レベル ($0 \text{ dBm} = 1 \text{ mW} @ 50 \Omega$) とも大きな差異が生じている。 $d = 2 \text{ mm}$ の空間では電界成分中に測定限界 1.8 GHz を越える分布が認められるのに対し、 $d = 40 \text{ mm}$ の空間では 1 GHz 未満の低域に多くのスペクトラムが集中し、 1 GHz を越えるスペクトラムの多くは受信電力レベルがスペクトラム・アナライザの測定限界(ノイズフロア: 約 -68 dBm) 以下になってしまっている。

すなわちスパーク放電によって導体の近辺に生じるインパルス性の電磁界は、空間をわずか 40 mm 伝搬することにより、その波形の鋭さを急激に失ってしまい周波数の低い連続振動成分(この条件下では約 80 MHz) が顕著に現われてくるようになる。

この傾向は物体からの距離にのみよるものではなく、帯電物体の電圧 (V_0) の高低

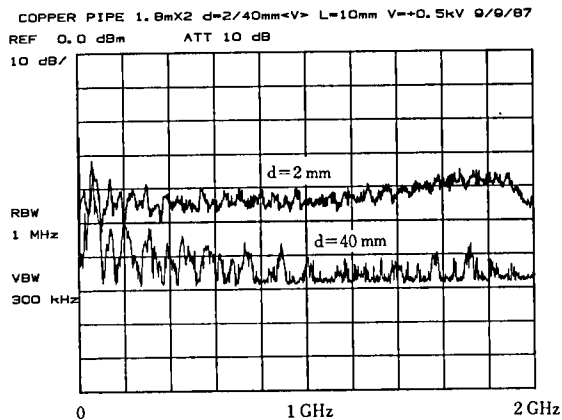


図 9 距離によるスペクトラムの違い

Fig.9 Spectrum differences with distance

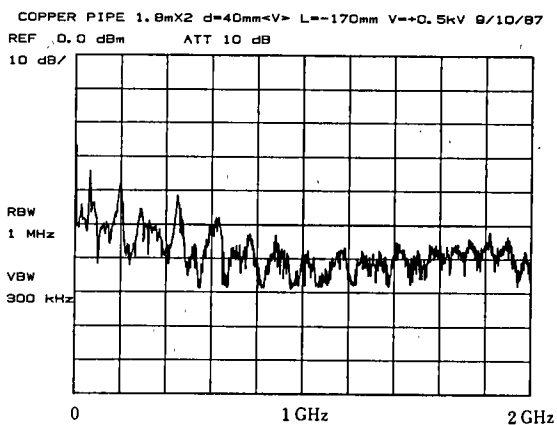


図 10(a) インパルス電磁界の周波数成分 $V_0=0.5$ kV
 Fig. 10(a) Electromagnetic spectrum vs discharge voltage, $V_0=0.5$ kV

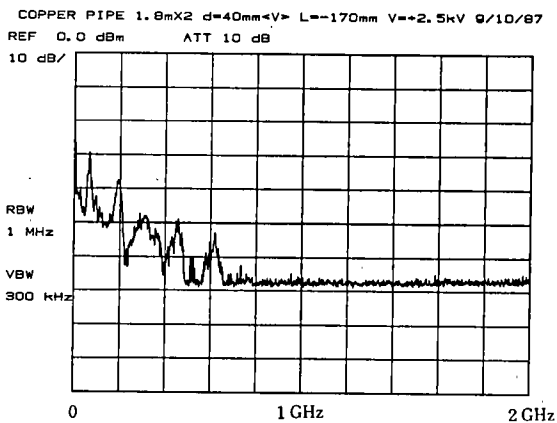


図 10(b) インパルス電磁界の周波数成分 $V_0=2.5$ kV
 Fig. 10(b) Electromagnetic spectrum vs discharge voltage, $V_0=2.5$ kV

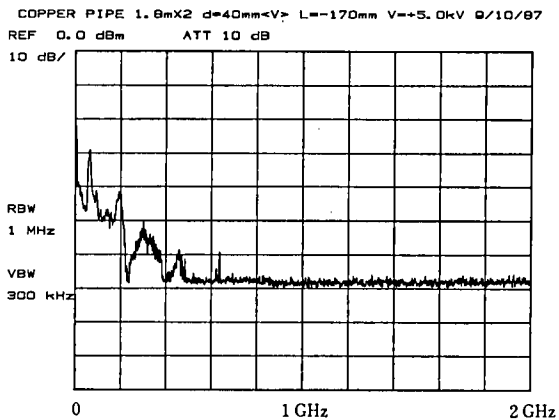


図 10(c) インパルス電磁界の周波数成分 $V_0=5.0$ kV
 Fig. 10(c) Electromagnetic spectrum vs discharge voltage, $V_0=5.0$ kV

によっても影響を受ける。銅パイプ A, B が固定されている条件のもとでは、 V_0 が高まるにつれ空間に放射されるインパルス性電磁界の周波数成分は図 10 のごとく低域へ移行し、 $V_0=5\text{ kV}$ の状態では分布幅は 500 MHz 未満になってしまう。

3.3 帯電物体の接近と離反における誘導 ESD

極めて狭い間隙 (約 1 mm 以下) を部分的に有した、相対向する二つの金属物体 a, b に対し帯電した物体 c が急に接近したり急に遠ざかったとき、この物体 (a, b) 近辺に配置してある ESD 検出器^[9] が反応することがあり、a, b 間で明らかに放電が発生していることがわかる。このような動的な静電誘導による金属物体間での ESD (以下誘導 ESD と称する) に関し、いくつかの基本的実験を行った。

3.3.1 帯電物体の動きと極性の反転

図 11 に示すごとく銅箔 a (長さ 500 mm, 幅 25 mm, 静電容量 $C_a=10\text{ PF}$) と、銅箔 b (長さ 100 mm, 幅 25 mm, $C_b=2\text{ PF}$) を約 0.1 mm の間隙を保ちアクリル板の上に直線状に配置する。銅箔 b から距離 40 mm の所に先の実験で使用したモノポール・アンテナを b と平行にセットし、同軸ケーブルでオシロスコープ (TEKTRONIX 7704 A) に接続する。放電の確認のため ESD 検出器をアクリル板から 500 mm の所にセットする。

銅箔 a の片端を指で触れ人体容量 $C_{HB} \approx 150\text{ PF}$ を通して接地する。⊖ に帯電してい

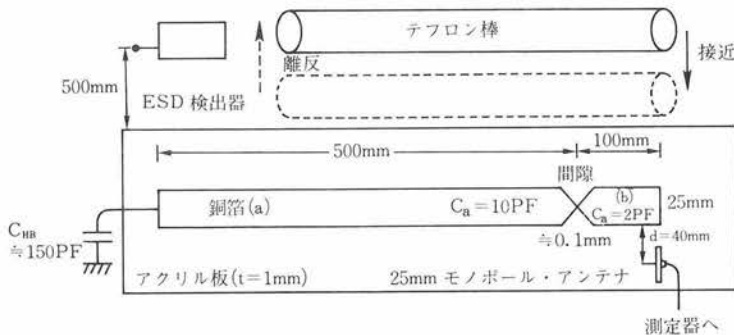


図 11 帯電物体の接近/離反における誘導 ESD の発生

Fig. 11 Generation of induced ESD as a charged body is brought near away

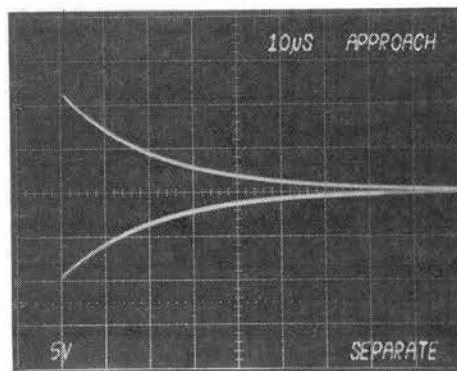


図 12 テフロン棒の接近と離反における誘導伝圧の極性の反転

Fig. 12 Polarity inversion as a teflon bar is brought near or away

るテフロン棒を約 500 mm 離れた所から近づけてくると、ある所で ESD 検出器が反応（受信レベル“LOW”）し a, b 間で誘導 ESD が発生したことがわかる。

ESD 検出器の反応の程度（“HIGH”または“LOW”）はテフロン棒の接近／離反時の速度の変化割合に対応する。

接近時の誘導 ESD によってモノポール・アンテナに誘起した電圧は、図 12 の上半分側のトレース（“接近”）に示すごとく放電に際してプラス方向への極性が現われ、数 10 μsec の時間を要して指数的に減衰する。

次にこのテフロン棒を急に遠ざけると放電が発生し、図 12 の下半分側のトレース（“離反”）に示すごとく接近時と逆の極性が現われ、同一の時間経過でゼロボルトにおちつく。テフロン棒が銅箔 a, b の近くにあつてこれが遠ざかる時、ESD 検出器はまったく反応しない。これは現行の ESD 検出器 (ES-81 V 型) の受信回路が正極性のインパルスのみをとらえる構造になっているためである。

誘導 ESD によって空間に生じる電界の大きさと、その極性は帯電物体の極性（正または負）と金属物体 a, b に対する運動（接近または離反）、そして速度の変化分の大小によって決定される。

3.3.2 偏波面の発生とその検証

ESD 検出器を用い誘導 ESD が発生したか否かを確認する際、この検出器のアンテナの向きによっては受信感度が著しく変化することがあり、これの原因を調べた。その結果、放電を引き起こした金属物体（波源）の長さ方向と検出器のアンテナの向きが一致すると、遠方（例：10 m）まで検出が可能であるが波源の向きと検出器のアンテナを直角にすると受信感度は著しく低下し、この場合は波源に近い所（例：2 m）まで遠くから ESD 検出器を移動しないと検知しないことがある。

すなわち ESD が発生した金属物体の形状によっては電界成分に関し、明確にしかも強い偏波面が空間に現われることがわかった。

この検証は ESD 検出器の存在と、その受信アンテナが極めて短いために初めて可能となったものである。

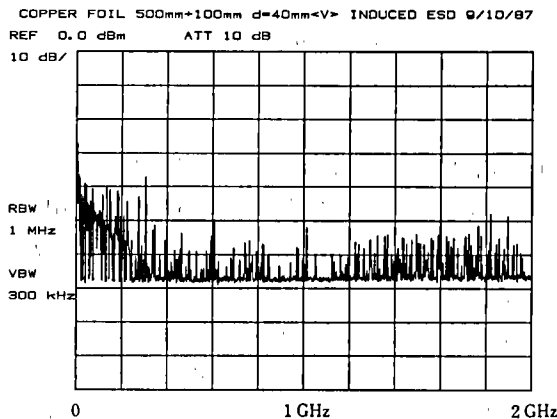


図 13 誘導 ESD のスペクトラム (500mm+100mm)

Fig. 13 Electromagnetic spectrum of induced ESD (500mm+100mm)

3.3.3 誘導 ESD による電磁波のスペクトラム

図 11 の構成において 25 mm モノポール・アンテナを同軸ケーブル (DGM 024-1000) でスペクトラムアナライザに接続し、テフロン棒の接近/離反による誘導 ESD のスペクトラムを調べた。その結果、図 13 に示すごとく測定器限界 1.8 GHz を越える周波数成分が認められる。

銅箔 a, b の長さを $a=b=25$ mm にすると先の例 ($a=500$ mm, $b=100$ mm) よりも強く、しかもより広帯域にわたる電磁波が放射される (図 14)。

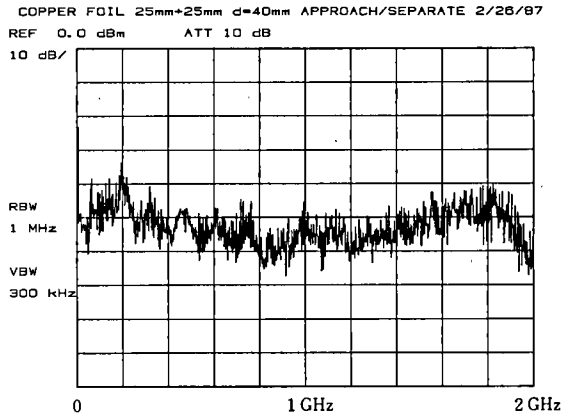


図 14 誘導 ESD による電磁波スペクトラム (250mm+250mm)

Fig. 14 Electromagnetic spectrum of induced ESD (250mm+250mm)

4. 検 討

4.1 ESD 試験結果を左右するもの

先に述べた実験 3.1 および 3.2 の結果を集約すると次のことが言えよう。

- 1) 自由空間中にある容量の小さい (数 10 PF 未満) 金属物体間での ESD は放電時に非常に損失が大きい。その値は、スパークに際しての電荷の転送量に注目すると約 45% が伝達されないで消耗している。

これに対しスパーク・ギャップを含む回路寸法が相対的に小さく、ほぼ集中定数回路とみなせる時のスパーク放電は損失が非常に小さく電荷の転送損失は 5% 未満である。

- 2) 放電点の前か後では物体 (導体) 近傍の電界の極性が反転する。この極性を持った電界の姿 (インパルス波形) は導体からの距離 (d) の差がわずか 40 mm 未満であっても、大きく変化し d が増大するにつれインパルス波形の鋭さが急激に失われてくる。

したがって、たとえ同一の ESD 試験器を用い同一の帯電エネルギー (電圧) を与える (受け取る) としても以上の条件の変化、ことに物体 (試験装置, 試験対象, 配線, 放電治具) 近傍の電磁界成分と試験対象とのかかわり合いが ESD 試験の都度変化するため、試験結果もこれらに左右され異なった Immunity Level が得られてしまうのである。

現行の ESD 耐性試験における再現性の悪さ (バラツキ) の根元を、以上のような観点から見直すべきである。

4.2 電磁エネルギーの加速度成分

静電エネルギー量 (多くは電圧表示) の大小と ESD 試験対象に与えるインパクトの大きさ (作用量) は、従来一対一で対応するものとされてきたが、実験 3.1~3.3 の結果に示すごとく、またわれわれが過去に行った一連の実験事実^{[1],[5],[7]} にもあるようにこの考え方には重大な疑問が生じる。

図 10 の一連のスペクトラムの変化が示すように、電極 (スパーク間隙部) が固定の状況での ESD に際しては電圧 V_0 と発生した電磁 (界) スペクトラム (f_{BW}) は反比例している。

さらに、またわれわれが行った別の実験によると図 15 に示すごとく電磁スペクトラム f_{BW} と受信電力レベル P_1 の積 $F: f_{BW} \times P_1$ [dBm·MHz] は、やはり V_0 に比例せず、ある所 (ここでは約 10 kV) でピークがあり、それ以降は V_0 が増大すれば $F(f_{BW} \times P_1)$ は低下の傾向を示す。

間接 ESD による電子機器への影響度合は、従来から何度か指摘してきたように物体の帯電電圧 V_0 に比例せず、ある電圧値で最も強い威力を発揮し、この時の電圧とインパクトとの関係は図 15 の V_0 と F の関係に一致することがわかっている。すなわち F 値が大きければ大きいほど ESD の威力が強いと言える。

ここで F 値を構成している単位としての [dBm·MHz] に注目すると、

- 1) 測定器にて表示される受信電力レベル P_1 は、入力アンテナがその場所で吸収

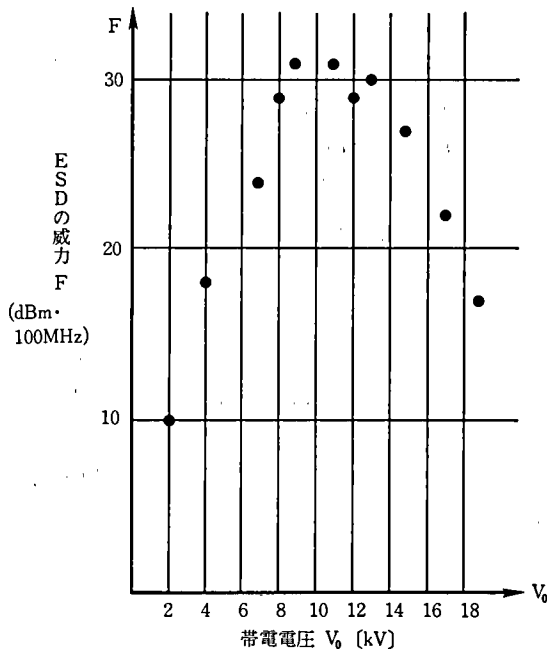


図 15 帯電電圧と ESD の威力 (F) の関係

Fig. 15 ESD power vs charge voltage

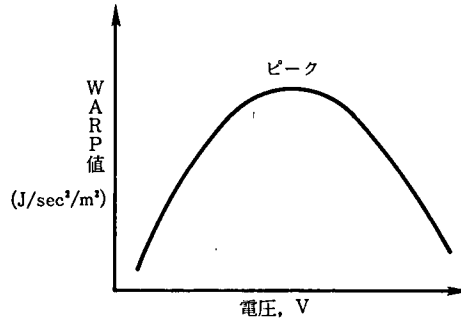


図 16 帯電電圧とインパルス EMI (WARP) の関係

Fig. 16 Impulsive EMI strength

した電磁エネルギー (電界 E と磁界 H の積: ポインティング・ベクトル) であるから, $P_1 = E \times H$ でありこの単位の次元は $[W/m^2]$ と示される。(電界 E の単位は V/m , 磁界 H の単位は A/m である。)

- 2) 電磁スペクトラム f_{BW} は周波数 [Hz] であるから, この単位の次元は $[1/sec]$ と示される。

したがって F 値の単位 $[dBm \cdot MHz]$ は,

$$F = f_{BW} \times P_1 [W/sec/m^2], \text{ または } [J/sec^2/m^2] \text{ と示される。}$$

$$\text{ここで, } W/sec \leftrightarrow J/sec^2 \quad 1W = 1V \cdot 1A$$

$$1J = 1V \times 1A \times 1sec \text{ である。}$$

すなわち F 値は ESD のインパクトの大きさを示すと共に, 物理的には単位の面積を通過する電磁エネルギーの加速度成分を現わしていることになる。

さらにまた, われわれがすでに 1984 年に提案していた概念^[1] (ESD によるインパルス EMI の強さを表示する) としての WARP 値 (Working Amplitude Rate of change Product); $WARP = V \times di/dt \times A$ も F 値とまったく同じ単位 $[J/sec^2/m^2]$ を有しており, この WARP 値は結論的にすべてのエレクトロニクスに与える ESD の衝撃的な作用力 (素子破壊・誤動作等) を統一的に説明できるものとする (図 16)。

5. おわりに

- 1) ESD 試験器で現実の ESD の世界をシミュレートするには, 電磁エネルギーの伝達に関し 1 次元の導体経路だけではなく, 3 次元の空間経路もあることを考慮しなければならない。
- 2) 静電スパーク放電が発生すると電荷損失が 45% 以上にも及ぶことがあり, 物体に蓄えられてきた静電エネルギーと ESD によって生じた電磁エネルギーとは 1:1 で対応しないと考えた方がよい。
- 3) 放電点の前後の空間は, 発生したインパルス電界の極性が正反対である。⊕に帯電していた物体の近傍は, ESD に際し Negative Going のパルス電界を, ⊖に帯電していた物体の近傍は Positive Going のパルス電界を作る。
- 4) 微小な間隙 (mm 未満) を有するゼロ電位にある二つの金属物体と帯電した物

体とが至近距離(約 500 mm 未満)にあり, 互いの間に相対運動(の変化)があると, この金属物体間で“誘導 ESD”が発生する可能性がある。

- 5) ESD の威力はエネルギー量ではなく, 単位の面積を通過する電磁エネルギーの加速度成分に比例する。そして, この威力は WARP 値によって示すことが最も合理的である。

- 参考文献 [1] M. Honda, T. Kawamura, “EMI Characteristics of ESD in a Small Air Gap”, ARP Governs the EMI—EOS/ESD Symposium Proceedings, EOS/ESD Association & ITRI, EOS-6, 1984, pp. 124~130. (日本語訳: “狭間隙状態における静電気放電の電磁妨害特性—ARP が EMI を決定する”, 日本ユニバック技報, 第 9 号, 1985, pp. 32~43).
- [2] W. Roth, P. Guest, G. Elbe and B. Lewis, “Heat Generation by Electric Sparks and Rate of Heat Loss to the Spark Electrodes”, The Journal of Chemical Physics, Vol. 19, No. 12, Dec. 1951, pp. 1530~1535.
- [3] 労働省産業安全研究所編, 「静電気安全指針」, 産業安全研究所技術指針, RIIS-TR-78-1, 10 月 1978.
- [4] H. Ott, “Noise Reduction Techniques in Electronic Systems”, (実践ノイズ通減技法) 松井孚夫訳 P. 27, 日本技術経済センター出版部 (JATEC), 1978.
- [5] 本田昌實, 小倉義美, 「ESD の特徴と計算機に対する影響(その 3)」電子通信学会技術研究報告〔環境電磁工学〕EMCJ 85-38, Vol. 85, No. 86, 1985, pp. 39~42.
- [6] IEC・TC65・Publication 801-2, “Electrostatic Discharge Requirements”, CEI, 1984.
- [7] M. Honda, Y. Ogura “Electrostatic Spark Discharges”, EOS/ESD Symposium Proceedings, EOS/ESD Association & ITRI EOS-7 1985, pp. 149~154. (日本語訳: “静電気スパーク放電による電磁妨害作用の考察”, 日本ユニバック技報, 第 13 号, 1987, pp. 3~12).
- [8] M. Honda, “A New Threat—EMI Effect by Indirect ESD on Electronic Equipment”, IEEE IAS Transactions Paper IUSD-87-92, Industrial Applications Society, 1987.
- [9] M. Honda, “Transient Electromagnetic Field Detector”, US PAT. No. 4, 631, 473 Dec. 23, 1986 (日本ユニバック・社有特許).

執筆者紹介 本田昌實 (Masamitsu Honda)

昭和 18 年生。40 年北海道大学 工教 電気科卒業。同年日本ユニシス(株)入社, 大型計算機の保守に従事。その後, 稼働環境, とくに計算機に与える電磁妨害, 静電気障害対策の調査・研究に従事。現在, プロダクト・サポート推進部サポート・エンジニアリング課所属。

静電気研究懇談会会員, 静電気学会会員, 電子情報通信学会会員, IEEE 会員, 米国電子戦学会 (AOC) 会員, 米国 EOS/ESD 協会会員。



LINC 適用によるソフトウェア開発事例

A Case of Software Implementation Using LINC II

平山道彦

要約 第四代言語 LINC を適用した情報処理システムの開発事例は数多くなってきた。LINC の適用は、少ない工数で高品質なシステムの開発を可能にしている。しかし、そのほとんどがソフトウェアの開発工程の下流部分（プログラム開発工程等）におけるものである。

今後は、開発工程の上流（分析・要求定義・設計）部分での高品質・高生産性の実現を目指したい。現在開発が進行中であるが、大規模・戦略的総合情報システム（システム規模は COBOL 換算で 200 万ステップ程度）の構築に関与する機会を得た。このシステムの開発に当たり、現状調査からプログラム作成までのすべての段階で、大幅工程削減によって高品質なシステムを構築するため、従来と異なる手法を試みている。

ソフトウェア開発工程の上流はデータフロー・ダイアグラム（以下 DFD）¹⁾ を利用しての要求モデルの作成とし、下流は、その DFD によって作成された要求モデルをインプットとし、LINC によってプログラムを生成し、プロトタイプを繰り返しながらソフトウェアを成長させ、完成させる方針である。

現在この手法により、高品質なシステムを高生産性によって実現すべく作業を行っている最中である。そのため公表できない部分も多々あることから記述にわかりにくい部分が存在するが、今後開発作業が完了し手法として整備が終了した時点で、機会があれば改めて報告する。ここに現時点までの開発状況を記述する。

Abstract There have been more and more information systems developed using the fourth-generation language, "LINC".

This means that "LINC" helps a lot bring forth a sharp reduction in the number of steps required for systems development while aiming to build up high quality systems.

However, most of such step cutbacks are seen only in the lower phase (program generation and the like) of software development process. Our future goal is to attain high quality and high productivity in the upper phase where systems analyses, more intensive definitions of user requirements and designing are included.

Under such circumstances we have recently been requested to reconstruct a nation-wide on-line realtime system, which requires about 2,000,000 steps if COBOL is used. The purpose of this work now under way is to implement a large-scale integrated information system as an enhanced strategic information system.

Taking advantage of this good opportunity, we have been attempting to apply new, non-traditional development methodology for higher quality through the sharply reduced number of steps in the entire software development process.

Our current idea of implementing the system is that we will produce required models by utilizing the data flow diagram (hereafter referred to as DFD) for the upper phase, while letting "LINC" generate programs based on the input of required models created by DFD for the lower phase so the software can

grow and be completed by repeatedly using plot-type paradigms, and so the computer algorithm can be generated with the help of the fourth-generation language, "LINC".

This paper outlines the new software development methodology as an actual development sample.

1. はじめに

今回紹介するシステム開発例は流通企業のシステムである。一般的にも言えるが、とくに流通企業でのコンピュータ化の特徴は、

- 1) ソフトウェア規模によって、必要とされる開発期間とは無関係に外的要因に依存して、ソフトウェアの最終成果物の納期が設定される
- 2) ソフトウェア開発の上流工程では、要求者が属する企業および業界の業務特性についての十分な知識を製作者が要求される
- 3) さらに、ソフトウェアの要求者（ソフトウェア適用による受益者）の生成されるソフトウェアに対する解釈と、ソフトウェアの製作者（ソフトウェア開発担当者）の生成されるソフトウェアに対する解釈とのギャップが、極めて生じやすい環境にある

ことである。今回のソフトウェア開発においても上記3点について十分に配慮し、いくつかの手法を組み合わせ使用している。説明の都合上、今回組み立てた開発手法を、一般的によく使用されているウォーターフォール・モデルとの対比で特徴的なところを記述する。またウォーターフォール・モデルとしては、ユニシスの開発手法である UNISYS Software Implementation Methodology（以下 SIM^[6]と略す）を例として使用した。

2. ソフトウェア・インプリメンテーション・ライフ・サイクル (SILC)^[6]

2.1 今回のソフトウェア開発手法とウォーターフォール・モデルとの対比

ユニシスの「SIM」(ソフトウェア開発方法論—ウォーターフォール・モデル)で設定している SILC^[6]は、①要求定義段階、②概要設計段階、③詳細設計段階、④開発段階、⑤導入段階、⑥評価段階、の6段階からなる。SIMでは各段階ごとの機能を独立させて各段階での成果物が設定され、各段階での中間成果物を生成した後に次の段階へ移行する。

この中間成果物生成の完了・未完了をもってソフトウェア開発の進捗を測定し、システム品質を維持する方式である。

SIMにおけるソフトウェア開発では、各段階が個々に独立し各段階ごとにその段階が必要とする専門家（業務専門家、コンピュータ技術専門家）をアサインし、個々の段階での高品質と高生産性を実現させる方法である。各段階は個々に独立しているため、各段階を完結すると中間成果物を出力し、次の段階はその中間成果物を入力として開発を開始する。この中間成果物の品質が最終目的の成果物の品質を大きく左右する。とくに各段階での最終成果物に対する解釈が、正確で統一した解釈を維持できるかが重要なポイントになる。

SIMでは図1で示すように問題領域から機械領域へ進行し、具体的レベルから要求モデルを作成する要求定義の段階、ビジネス・モデルを作成する概要設計段階と抽象

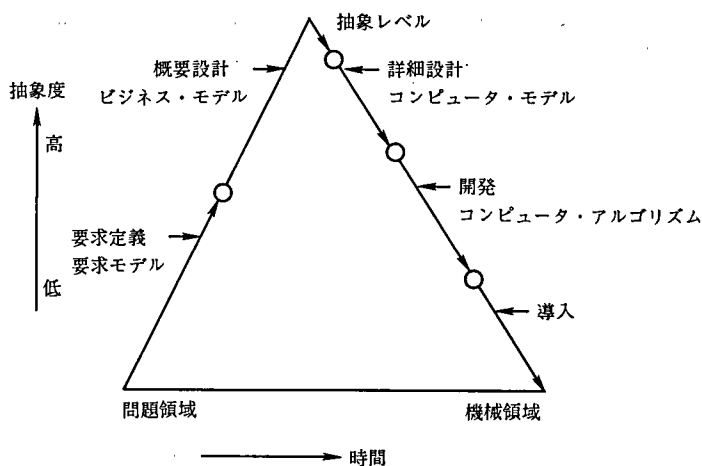


図 1 問題領域から機械領域への推進^[6]

Fig.1 Mountaine model of software implementation

的レベルになり、コンピュータ・モデルを作成する詳細設計段階、コンピュータ・アルゴリズムを作成する開発段階、導入段階へと具体レベルへ戻る。問題領域から機械領域への各段階で、次段階のために作成される中間成果物をもって、正確にインタフェースするためには多くの工数を必要とする。

さらにこの中間成果物は時間の経過によって発生する変化に対応し、精度を維持するためにも多くの工数が必要である。また、この中間成果物から操作的仕様^[4](業務運用オペレーション遂行のためのドキュメント類)の流用も困難である。

今回の開発手法では、

- 1) 中間成果物はできるだけ削除する
- 2) 最終成果物(生成ソフトウェア)を各段階で成長させる。すなわち、ビジネスモデル即ソフトウェア生成となるようにする
- 3) 操作的仕様は最終成果物である生成ソフトウェアに内蔵し、必要時自動生成させる

ことにより、ウォーターフォール・モデルで必要とされる中間成果物作成工数、操作的仕様作成工数を大幅に削減し、システム要求者およびシステム受益者と製作者とのギャップから生ずるシステム品質の低下を防止する。

第四代言語 LINC は機械領域での設計段階を必要とせず、ビジネスモデルをそのまま入力とすることが可能なことから、モデル作成が実現モデルより要求モデルに近いモデルを唯一構築し、そのモデルを成長させていくことができる。

2.2 プロジェクト推進モデル

一般的にソフトウェア開発で使用されるモデル^[1]としては、

- 1) プロジェクトを推進するためのモデル
- 2) 要求仕様を表現するためのモデル
- 3) 機械設計を表現するためのモデル

の3種類が存在する。今回のソフトウェア開発の開発過程でのドキュメントは要求仕

様モデルのみ作成した。機械設計モデルは、第四世代言語 LINC 機能を利用することで一切作成していない。

一方、開発のプロジェクトを推進するためのモデルを作成し開発を制御した。これらの要求仕様モデルとプロジェクト推進モデルは共にDFD, データ・ディクショナリ^[4] (以下 DD) 等を使用してモデルを表現した。

今回の開発例を図2のプロジェクト推進モデルによって説明する。

基本的に開発工程で作成される成果物は、

- 1) 形式的仕様^[2]としての要求仕様モデル
- 2) 要求仕様モデルから転換する LINC 入力スペック
- 3) 操作的仕様が自動生成されるためのスペック

からなる。今回のシステム開発に当たって、形式的仕様としての要求仕様モデル (一部は設計モデルを含む) の記述には DFD と DD を用い、さらに DFD 上で業務機能・業務活動の一かたまりを表現するバブル^[3]の補足が必要であれば、バブルに対応してミニ仕様書^[3]を作成した。

モデル作成に DFD を採用した理由は、

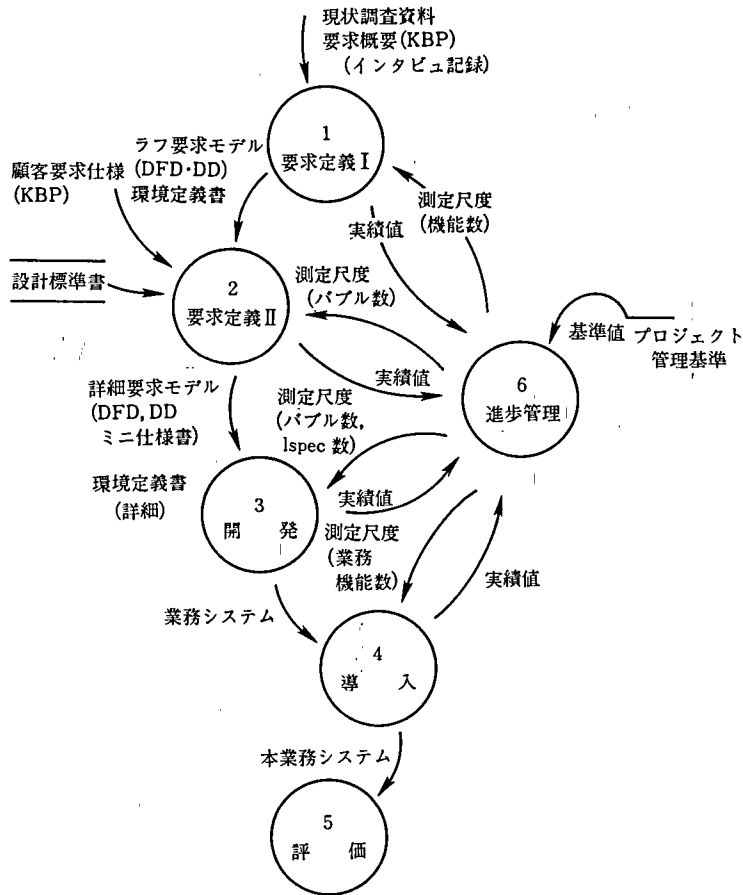


図2 プロジェクト推進モデル (開発フレームワーク)
Fig.2 Phase network (implementation frame work)

- 1) 要求者の解釈(モデル)と製作者(実際に開発を担当するシステム・エンジニア)の解釈を一体化し、より要求者寄りで要求者が十分に理解できること、要求者自身がモデル作成に直接参画できること、さらにできあがったソフトウェアの検証にも利用可能なモデル記述ができること
- 2) 要求仕様モデルが LINC のソフトウェアを生成するための入力スペック (具体的には、1 または複数のバブルが LINC の入力の単位 (以下 ISPEC⁽⁷⁾) とほぼ一致すること
- 3) 各段階で実施されるプロトタイプ時に、エンドユーザ (システム受益者) の理解を容易にさせるためのマニュアルとなり、プロトタイプ実施と同期して成長させる (バブルを階層的にブレイクダウンさせる) こと

などがあげられる。

プロジェクトを推進するための組織および役割は、プロジェクトの発足時から終了まで固定されているものと、プロジェクトの進行に伴って変化するものとに分割した。固定的なものとしては、資源の確保 (必要スキルを有するマンパワーの調達と、時間的に可能であれば、要員育成のための教育計画の作成および教育の実施)、品質の制御、技術的コントロール等であり、プロジェクト内の個人またはグループに分担させた。

流動的なものとしては開発作業そのものであり、図3に示すような組織で推進している。今回の開発に当たっての基本的な原則は、「ソフトウェア開発は要求定義から開発・評価の段階まで同一のグループが担当する」こととし、開発は業務別のチームを編成し、開発工程別の分割は一切行わないことにした。

実際の状況としては、図3のようにチームはかなり各段階で集散する結果となった。

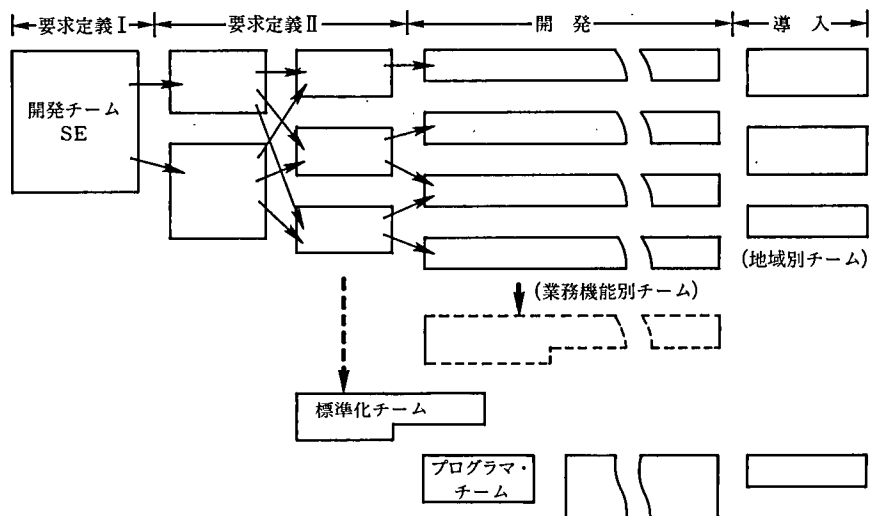


図3 開発チームの集散状況

Fig.3 Implementation teams

• ISPEC: ビジネス・モデルと LINC の領域をインタフェースする処理単位 (主にオンライン処理系)

1 チームが1業務機能を担当し、モデル作成からソフトウェア生成までを可能とするには、初期の過程で企業全体の業務機能を理解することが必要となる。したがって開発の進行に伴い、開発メンバが業務機能を理解する度合を見ながらチームを徐々に分割していくことになった。そのため作成される要求仕様モデルもチームの分割ステップと同期して詳細度合が増すことになる。併行してグローバルな分野については別の開発チームを設定し、業務機能間（担当チーム間）インタフェースの品質を保持するために各チームから抽出したメンバによる検討チームを設定した。

プロジェクトの進捗を管理する方式としては、SIM で設定した成果物ベースとは異なる方式を用いた。各段階ごとに図2のごとく測定尺度を設定した。しかし今回は、過去の実施例が存在しないことから進捗測定をする際の基準値がないため、各段階では当初当て推量とし、各段階内の作業の進行に伴って抽出された実績値を予測子とし予測値を算出した。

測定尺度としては図2のごとく、

- 1) 要求定義Ⅰ段階（ラフ要求モデル作成段階）……ユニシスのシステム化計画手法（Knowledge Based Planning, 以下 KBP）で提供されるテンプレート *^[5] で示された業務機能・業務活動に対してモデル化完了の業務機能・業務活動数によって進捗を管理
- 2) 要求定義Ⅱ段階（詳細要求モデル作成段階）……DFD によって作成されたラフ要求モデルに存在する総バブル数に対して、詳細化されたバブル数により進捗を管理
- 3) 開発段階……DFD によって作成された詳細要求モデルに存在する総バブル数に対して、LINC の ISPEC 作成が完了したバブル数により進捗を管理

今回の開発実施例から各段階での測定尺度の実績値は、今後の同種の開発手法による開発の基準値作成に貢献できるものと考えている。

3. 要求定義Ⅰ段階（ラフ要求モデルの作成）

第一の段階ではラフな要求モデルを作成した。この段階では、できるだけ開発メンバ全員が一つの検討チームで作業することを基本とした。しかし、生産性を上げるために一部は機能別にチームを編成してモデルを作成せざるを得なかったが、かならず開発メンバ全員による検証を行い、全員の業務システムについて理解の度合を一致させることが必要である。

この段階の成果物は要求モデルの幹を作成することである。そのためシステム全体を理解し、最終成果物としてのソフトウェアの適用範囲・機能レベル・管理レベル等を明確にしなければならない。今回の開発例では、この段階での成果物は具体的に企業の全体モデルであることから、企業の主要業務機能を10バブル程度で構成させた。つぎに個々の業務機能を業務活動にブレークダウンし、各業務機能別に10バブル程度で表現したモデルを作成した。

以上の全体モデル1ページ10バブルと、業務機能別約10ページ100バブルで構成されたモデルをラフ要求モデルとした。バブル間のインタフェースについても必ず記

• テンプレート：業種別の標準的企業内の業務機能・業務活動一覧表

述し、インタフェースの表現はデータのブロック単位(たとえば“出荷指図伝票”, “取引先台帳”等)程度にとどめ、インタフェース名をDDとして作成した。

第一段階での要求定義 I のプロジェクト推進モデルは図 4 となる。今回のシステム開発例は企業全体を対象としていることから、企業全体としてシステムの高品質(業務の重要度、今後の企業戦略との整合性等)を確立するために、システム化計画のツールとしてユニシスの KBP を適用している。

まず業界汎用のテンプレートを今回の対象企業の扱い品種、販売チャネル等から組織活動別にブレイクダウンし、対象企業独自のテンプレートを作成した。つぎに企業レベルの CSF (Critical Success Factor, 最重要成功要因⁶⁾) を考慮しながら要求仕様および解決策記述書を作成した。形式としては、要求に対する処理、処理に必要な入力、要求に対応する出力の 3 項目によって記述する。

これら今回の対象システムの特異性を加味したテンプレートと、顧客の要求仕様記述書を現状のモデル (DFD, DD により記述) に追加・修正し、ラフ要求モデルを作成

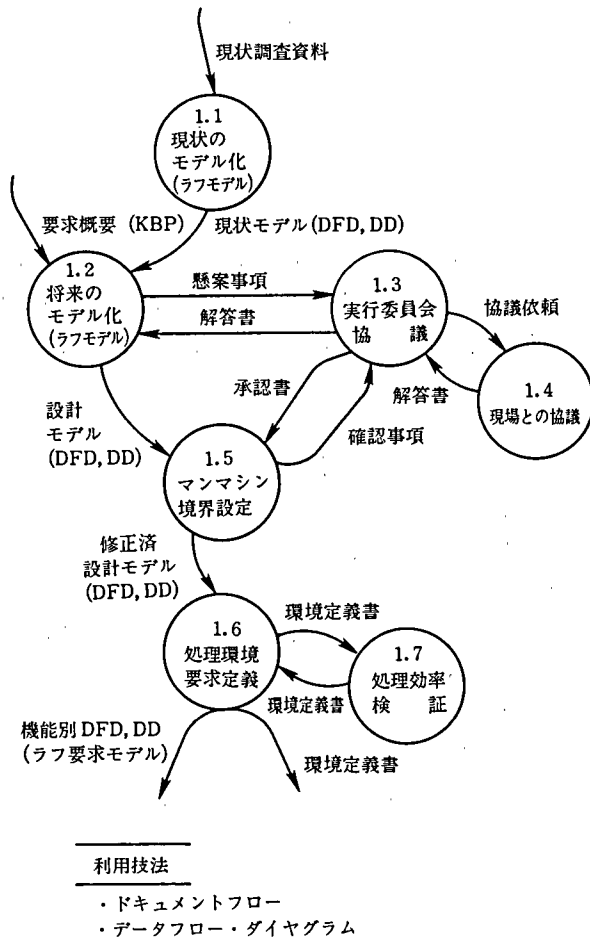


図 4 プロジェクト推進モデル (要求定義 I)

Fig. 4 Phase network (requirements definition phase I)

していった。作成されたモデルの正統性を検証するためにエンドユーザ部門により構成される検討チームを設置し、検証作業とモデル作成時に発生する種々の懸案の解決を行った。

要求定義Ⅰの段階では、将来モデルのDFDとDD記述の中にデータ発生状況(データ発生量・発生サイクル・管理レベル等)を追加記述し、データ発生状況を基に機械領域での設計技術者による環境定義書(機器設定・ネットワーク設計書等)を作成した。

要求定義Ⅰ段階を実施した結果の重要な点は、

- 1) モデル構築に当たり業務機能別・地域別から適切な要求者を選定し、モデル作成への参画計画、ならびにタイムリなインタビュ実施計画、要求モデルのレビュー実施計画を綿密に立案せねばならない。
- 2) 開発メンバ全員がモデル化のツールであるDFD, DD, ミニ仕様書を使いこなせるように構造化分析技術の修得方法, OJTの方法, ウォークスルの有効な進め方を初期の段階で確立しなければならない。注意しなければならないのは、構造化分析手法による要求モデル作成には初期の過程で非常に多くの工数がかかることと、多くの工数をかけて徹底的な検討を体験しなければ、開発メンバのモデル作成能力は向上しないということである。そのため、初期過程の徹底度合が全体のモデルの品質を左右することになる。

4. 要求定義Ⅱ段階(詳細要求モデル作成)

要求定義Ⅱでは、要求定義Ⅰ段階で作成したラフ要求モデル(企業全体のモデル、業務機能別のモデル、データのブロック単位記述によるDD)を詳細化(ブレイクダウン)することである。モデルを詳細化するだけではなく現行の組織、今後予想される組織、地域性等が加味されることにより業務機能のモデルを現実的モデルへと成長させる。DDについてもデータをアイテムレベルに落とす必要がある。この要求定義Ⅱの段階のプロジェクト推進モデルは図5のようになる。

要求定義Ⅱ段階の初期には、詳細モデルを記述するために標準書(既存の標準書を基礎に修正する)を作成する。作成に当たっては、次の開発段階も考慮する必要がある。なぜならば開発段階に入る以前に、1あるいは複数個のビジネスユニットが具体的に受益者の運用を可能にするモデルになっていなければならない。したがって、完成したモデルが机上で稼働しなければならない。この時点で曖昧さを残すと、開発段階で作成されるISPECとのギャップが生じ、プロトタイプ実施時に多くの不具合修正が発生する結果となる。

要求定義Ⅰで作成された業務機能別に10個程度のバブルが定義されているが、具体的にはその1バブルを最大10バブルにブレイクダウンする。今回の開発例では、DFDのバブルを単純にブレイクダウンすることができなかった。ラフ要求モデルは、企業として必要とされる業務機能(テンプレートが基礎)をもとに網羅的で汎用的モデルになっているが、詳細要求モデルではラフ要求モデルの1バブルが企業組織別・地域別等から同一レベルで分割されるケースが非常に多くなったことと、ブレイクダウンの深さを事前に3~4レベル程度に留めることを標準書に明記しなかったことから、モ

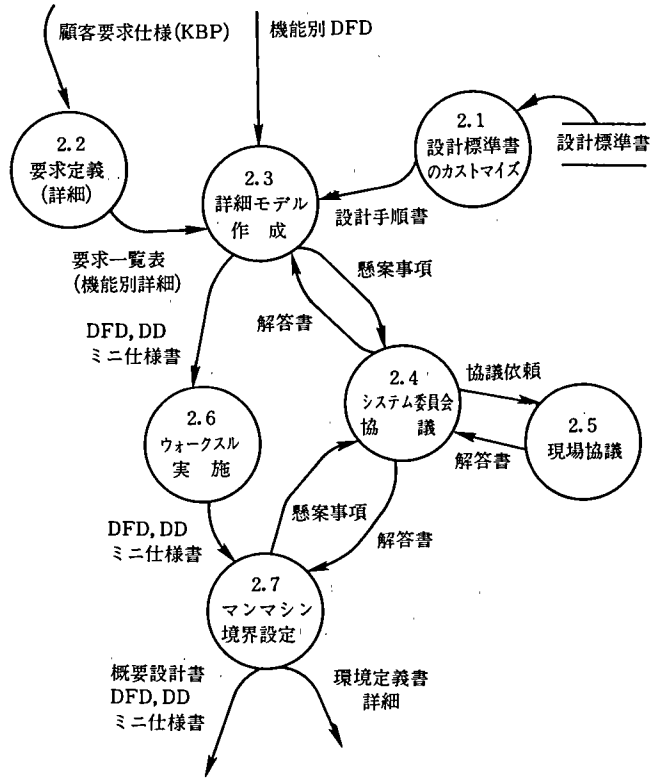


図 5 プロジェクト推進モデル (要求定義II)

Fig.5 Phase network (requirement definition phase II)

デル化の意味¹¹⁾(ある目的を持った論理表現を解釈し、その真偽を論ずるために設定する概念の記述や図式)とは異なり、理解しにくい記述に陥ることになった。なぜならば、システム要求者および受益者を含めてモデルの詳細化を実施する場合、ラフ要求モデルの一つのバブルを垂直的にブレイクダウンする傾向があるが、バブルを1階層ブレイクダウンした場合は、その階層すべてを完結させながら進めなければならない。モデルの階層は3~4レベル程度に限定すべきであり、4レベル目のバブルが最少化しない場合はバブルの説明のためにミニ仕様書を作成した。

DFDのインタフェースについては、すべてのアイテムレベルに分解してDDを作成する。要求定義II段階でのDDのアイテムレベルの記述は、次の開発段階の初期に実施するデータのシノニウム削除を可能にするためにアイテム名の統一を図りながら作成しなければならない。今回の開発例では、シノニウム削除を要求定義IIの段階ではほとんど考慮せずに進めたため、統一化に多くの工数を割かざるを得なかった。

要求定義IIの段階での実施結果として、いくつかの注意すべき点がある。

- 1) あくまでもビジネス・モデルをDFDの使用により構造化記述しなければならないが、一つの業務機能または業務活動がDFD上で階層化して記述できるとは限らない。システム要求者とのギャップを減少させるためには、同一レベルのバブルが羅列されるケースもある。

2) 要求定義IIの段階に入ると、実際の詳細なビジネスのモデル化を進めるうちに製作者(SEグループ)によっては、コンピュータ・アルゴリズムの設計上からモデル化する傾向が出てきてしまう。とくにインタフェースおよびデータの集合体に関するデータ記述が、ビジネスのイベントによる集合ではなく、コンピュータ処理の便宜のための集合になる傾向がある。

しかし今回の開発例では、要求定義I・IIで作成した要求仕様モデルが唯一の成果物となっていることから、設計モデル的(機械化モデル)な要素を加味する必要が出た。さらに機械化モデル寄りの設計を追加した場合でも、ビジネスイベントを考慮した上でないとシステム品質が保障されなくなる。そのための要求者寄りのモデル作成であることを作成過程で徹底せねばならない。

5. 開発段階

開発段階ではLINCを使用し、前段階(要求定義II)で出力されたDFD, DDおよびミニ仕様書により作成されたモデルを入力とし、目的とするソフトウェアおよびそのシステムと同期した操作的仕様書等のドキュメントを生成する。

開発段階には、マイルストーンとして2回のプロトタイプの実行がある。基本的には、これらのプロトタイプ^[2]は開発の進行によってソフトウェアが成長し、最終的に目的のプログラムが生成されるパラダイム^[4]である。今回の開発方式でも、要求定義IIの段階と開発の段階の機能は問題領域と機械領域^[6]に分割されているが、要求定義IIのレベルで作成された要求モデルがそのまま成長し具現化される。

5.1 プロトタイプI

プロトタイプIは、開発メンバに対して開発の作業を進めていく上での技術的根拠を得るためのOJTとなる。そのための準備工程として標準作業手順書を詳細に分析し、今回の開発での特異性を加味して作業手順書を作成する。その手順書に添って要求定義IIで作成された詳細要求モデル(DFD, DD, ミニ仕様書より構成)からサンプルを抽出し、開発から導入までの全工程を実施し、サンプル・ソフトウェアを生成してみる。このサンプル・ソフトウェアを生成するまでの過程から発生した不具合を検討し、作業手順書の改良に結び付ける。

プロトタイプIで実施されるOJTの内容は、図6(プロジェクト推進モデル(開発段階))で示される「プロトタイプIの実施」(バブルナンバ4.3)の次のバブル(4.4~4.14)に記述されているジョブのすべてが対象となる。

基本的なプロセスは次の通りである。

- 1) バブル(4.4): ISPEC設計……DFDのバブルを処理単位にくくることによりISPECを決定し、一覧表(図7)を作成する。
- 2) バブル(4.6): データの正規化……DDからシノニウムを削除し、ISPEC(ビジネスのイベント単位)でデータの正規化(データの集合化)を図る。
- 3) バブル(4.10): データ辞書の作成……DDからシノニウムを削除し、システム全体(今回は企業全体)でのデータアイテムをすべてLINCのデータ辞書に登録する。
- 4) バブル(4.11): 画面登録……LINCのスクリーンペインタにより画面登録を実

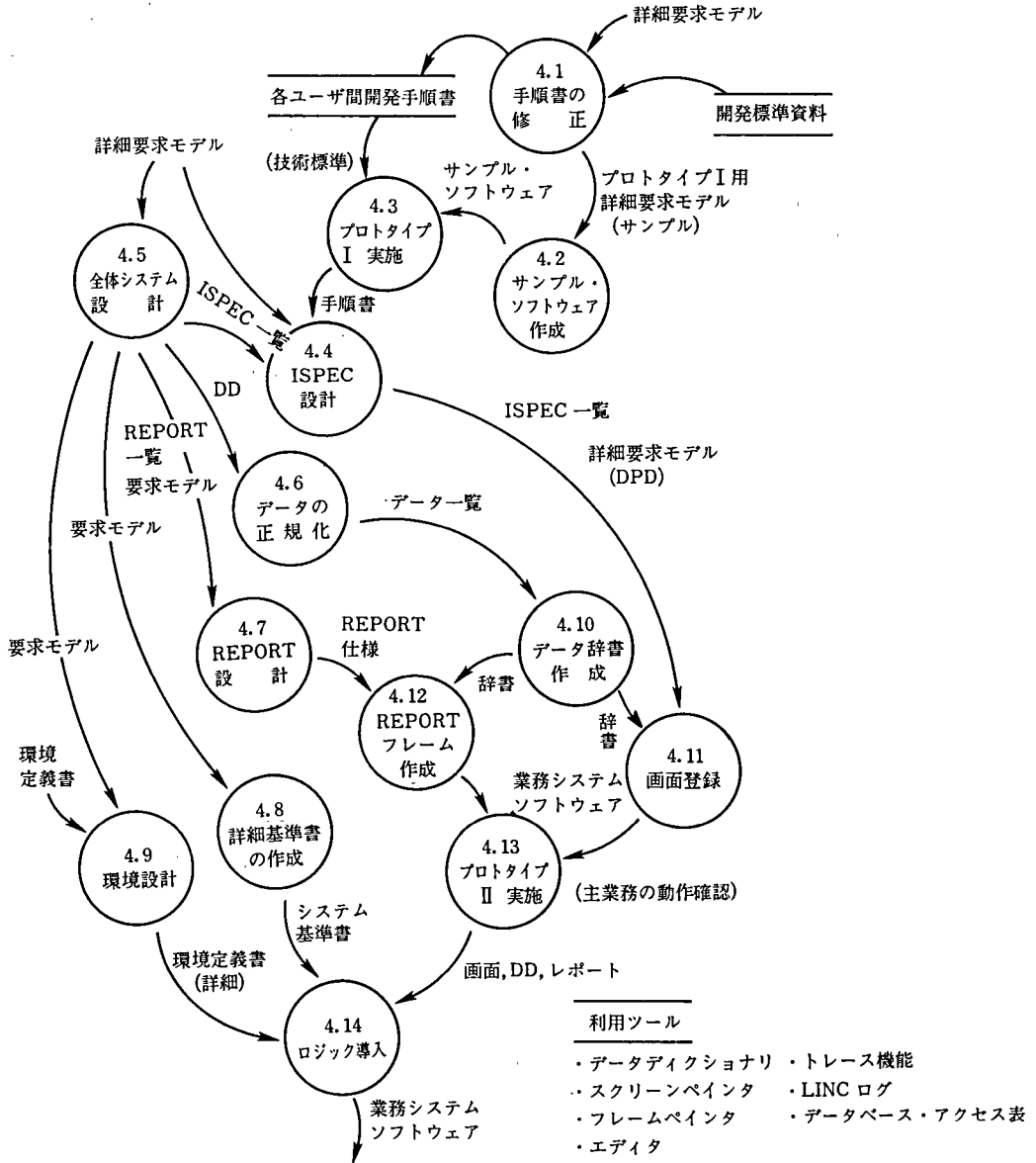


図 6 プロジェクト推進モデル (開発段階)
 Fig. 6 Phase network (development phase)

施。この登録はただ単に画面に登録したのではなく、業務ロジックのないソフトウェアが生成されシステムとして稼働する。

5) パブル(4.14)：業務処理ロジックの導入……画面登録により生成されたソフトウェアに業務処理ロジックを導入する。

さらに必要とされる帳表等の出力関連は、LINCのフレームペインタによりイメージに登録し、さらに業務処理ロジックの導入を行う。

以上のような過程でプロトタイプIを実施した。抽出するサンプルは3個程度のバ

LINC ISPEC/REPORT 一覧表

機能 コード: 名称:	タイプ C: 標準コンポーネント M: メモコンポーネント E: イベント R: レポート 入出力区分 I: 入力 O: 出力 I/O: 入出力 難易度 A: 難 B: 管 C: 易	ページ: / 作成日: 平成 年 月 日 作成者:				
NO.	組織	ISPEC/REPORT 名 [英字名]	タイプ	区分	難易	DATA FLOW DIAGRAM NO.
		[]				
		[]				
		[]				
		[]				
		[]				
		[]				
		[]				
		[]				
		[]				
		[]				

図 7 ISPEC/REPORT 一覧表

Fig. 7 Example ISPEC/REPORT work paper

ブルを 1 ISPEC とすることが適当である。さらに 2 個程度のバブルを 1 REPORT*^[7] として設定し、ソフトウェアを生成する。この 2 種類の開発方法を理解するための OJT は約 10 日を要する。とくに重要なことは、この OJT により開発メンバ自身が開発手順を理解するとともに、今回のシステム開発に適用可能であるかを検証し、実際の開発段階で品質の高い進捗管理を実現するための測定尺度の正当性と、それに対する基準値を得ることである。

5.2 プロトタイプ II

プロトタイプ II は要求者または、システム適用による受益者が新システムのイメージを得て、新システムの業務プロセスを理解しさらにシステムの検証を実施する。

プロトタイプ II の実施までの基本プロセスは、

- 1) バブル (4.4) : ISPEC 設計
- 2) バブル (4.6) : データの正規化
- 3) バブル (4.10) : データ辞書の作成
- 4) バブル (4.11) : 画面登録

を作業手順書に添って実行する。開発の組織としては、要求定義 II で確立された業務処理別 (開発業務のサブシステム単位) チームが同一メンバで継続して作業する。しかし、REPORT 等の少量の記述で作成された中間成果物 (プリプリントされた帳表イメージと DD リストに少量の処理ロジックを記述した仕様書) でプログラマ・グループに委託可能なものは分離して開発した。さらにグローバルなデータの正規化、グローバルな業務処理ロジックを抽出し共通の ISPEC 開発チームを編成するなど、開発

* REPORT: ビジネスモデルと LINC 領域とのインタフェースをする処理単位 (主にバッチ処理系/出力処理系)

の生産性向上を考慮して進めている。

ISPEC 単位の画面登録、REPORT のフレーム作成が完了すると業務処理ロジックは入っていないが、稼働するソフトウェアが生成される。その間の作業としてはデータ辞書の登録と画面の登録作業のみであるから、非常に短期間でプロトタイプIIの実施が可能になる。しかし、プロトタイプIIの実行にはその目的を達成するまでにかなりの期間と工数を必要とした。

今回はプロトタイプII実施計画を作成しエンドユーザ部門の実務担当者、中間管理者、上級管理者の各レベルごとに内容を選別し実施したが、結果として全従業員の25%に及んだこと、実施には画面ハンドリングと帳表確認だけでなく未導入の業務処理ロジックを詳細要求モデルを利用して事前に確認すること、画面および帳表等のデータの管理レベル、発生量、発生サイクルなどを同時に調査したことから予想以上の期間と工数を必要とした。

今回の開発例におけるプロトタイプII実施後の考察として、

- 1) 要求定義IIでのDFDによるモデル作成時バブルの最少化を徹底し、 n 個のバブルと対応する1ISPECが完全に相似形になるようにしなければならない。
- 2) ソフトウェア上にはまだ業務処理ロジックが導入されていないが、プロトタイプIIを業務処理ロジックから実運用のレベルに至るすべてについて最終的な確認をすることから、システム要求者およびシステム受益者のシステム稼働前準備の開始を大幅に早めることが可能となり、本稼働をスムーズに実施することができる。
- 3) プロトタイプIIの実施相手は、システムの要求者および受益者であることからシステム製作者側からの一方的実施だけではなく、参画を促進するために一定期間常時検証可能な環境を設定することがシステム品質を向上させることになった。

以上のように、プロトタイプIIは開発段階での重要なマイルストーンであり、開発段階でのシステムの品質と同時に開発作業の工数を決定するポイントとなる。

プロトタイプIIの実施後、プロトタイプIIで確認された業務処理ロジックをソフトウェアに導入する。業務処理ロジックの導入作業は、要求定義IIの段階で作成されたDFDによるモデル(プロトタイプII実施時の修正済)により行う。したがって開発チーム編成は、変化させずにISPEC/REPORTの開発作業を継続させることでシステムの品質の維持と高生産性の確保をした。

6. 導入段階

導入段階ではデータ作成および投入、現場教育、併行処理が実施される。今現在、開発例はこの段階に入っていないことから実証による考察はできない。この段階にはマイルストーンとして2回のプロトタイプが実施される。

6.1 プロトタイプIII

開発段階からマイルストーンとしてのプロトタイプ実施を繰り返し、ソフトウェアを成長させてきたが、プロトタイプIIIでは業務処理ロジックが導入されてほぼ完成した最終成果物であるソフトウェアにより、システム要求者および受益者に検証させる

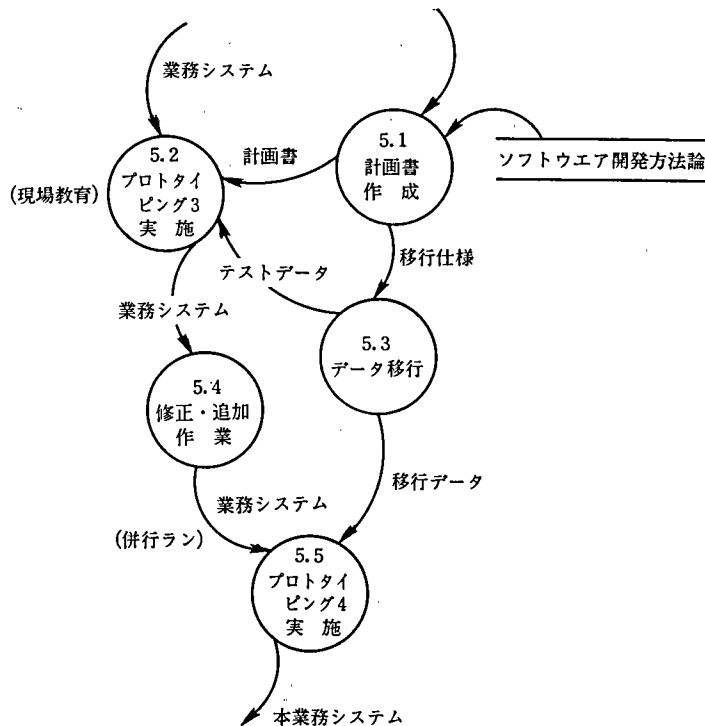


図 8 プロジェクト推進モデル (導入段階)

Fig. 8 Phase network (implementation phase)

ことになる。そのためにデータの準備，データベースへのデータの投入，トランザクション・データの収集を行い，より実業務に近い環境を設定する。さらにプロトタイプIIIは現場教育を目的として対象部門，対象者を選定する。また完成ソフトウェアから操作仕様書の自動出力，DFD，DDによる要求モデル等のドキュメントを準備し，要求者および受益者へのシステムの検証および教育に利用する。

6.2 プロトタイプIV

プロトタイプIIIで発生した改善項目を検討し，必要であればソフトウェア，DFDによる要求モデルの修正追加，操作仕様書の自動出力を行う。この時点でビジネスユニットとしてのISPECを機械処理の効率を考慮に入れた構造にて再構築する。最終的なマイルストーンとしてのプロトタイプIVを実施する。目的としては，要求者および受益者による最終検証と併行処理の実施である。実施は過去3回のプロトタイプと異なり，業務機能別の実施ではなく企業の組織・地域単位に実施しなければならない。

7. おわりに

ソフトウェア開発の全段階について一貫性を持たせて，ウォータフォール・モデルとは異なる手法による開発例をここに報告したが，個々には多くの問題が発生している。とくに，

- 1) 要求モデル作成技術が未熟なこと

- 2) バブルから ISPEC 移行の標準化が困難なこと
- 3) DD からのデータの正規化技術が未確立であること

といったものが今後の課題となった。

しかし今回の開発例では、システム全体としての品質について十分に維持できているのは6か月間に及ぶシステム化計画の作成段階(KBP 利用による)が存在したことによる。

今後、システム化計画における出力とも一貫性を持たせたシステム開発手法の確立を図る必要がある。

-
- 参考文献 [1] T. DeMarco, ソフトウェア開発プロジェクト技法, 近代科学社, 1987.
[2] 松本吉弘, ソフトウェアに対する要求の形成, 情報処理, 社団法人 情報処理学会, Vol.28, No.7, JULY, 1987.
[3] T. DeMarco, 構造化分析とシステム仕様, 日経マグローヒル社, 1987.
[4] 大野 豊, ソフトウェア工学の背景と展望, 情報処理, 社団法人 情報処理学会, Vol. 28, No.7, JULY, 1987.
[5] Knowledge Based Planning 解説書, 日本ユニシス(株).
[6] Software Implementation Methodology 解説書, 日本ユニシス(株).
[7] LINC II LDL 解説書, 日本ユニシス(株).

執筆者紹介 平山道彦 (Michihiko Hirayama)

昭和21年生, 45年東京理科大学理学部物理学科卒業。
同年日本ユニシス(株)に入社。製造工業および流通業関連
システムの設計・開発を担当。現在システム第二本部 シ
ステム統括二部 流通システム一部所属。



順序ソート代数に基づく代数的仕様と始代数存在判定

Algebraic Specification Based on Order Sorted Algebra and a Test for Existence of the Initial Algebra

清水 敏彦

要約 順序ソート代数に基づく代数的仕様記述法と、その始代数存在判定について報告する。順序ソート代数に基づく代数仕様は、従来の多ソート代数に基づく代数的仕様記述法を、ソートが示す集合の間に包含関係を定義できるように拡張したものである。

この方法によると、従来の多ソート代数に基づく方法では記述できなかった、集合間の包含関係、部分演算、演算の多種継承、多義演算を表せ記述能力が高い。しかし、ソート間の半順序が代数に新しい制約をおき、記述した仕様の意味を定める始代数が存在しないことが起こる。このため記述した仕様に始代数が存在するかを判定するために、存在の十分条件である正則性と接続性を調べる手続きを実現した。

Abstract Order-sorted algebra generalizes conventional many-sorted algebra by adding a partial order on the set of sorts. This generalization contributes to the theoretical treatment of abstract data types with partial operators, overloaded operators, multiple inheritance etc. However, a partial ordering also imposes an additional restriction on an algebra and this fact causes a non-existence of initial algebra for some specification written in the frame of order-sorted algebra theory. This paper gives a test procedure for checking the properties of 'regularity' and 'coherency' that are sufficient condition for the existence of an initial algebra.

1. はじめに

抽象データ型 (abstract data type) は、プログラミング方法論におけるモジュール化、情報隠蔽、抽象化、局所化などの原理を実現する有用な概念である。米国防総省が中心になって作成したプログラミング言語 Ada においては、この抽象データ型の概念はパッケージとして実現された。

抽象データ型の仕様の厳密な記述方法として代数的仕様記述法 (algebraic specification method) がある。代数的と呼ぶ理由は、抽象データ型を抽象代数学でいう代数として捉えて仕様記述するためである。仕様記述は公理と呼ぶ等式の集合で、演算の間に成立する関係を記述することにより行う。代数的仕様記述法には等号論理と始代数に基づく宣言の意味論や、等式を書き換え規則とする項書き換え系に基づく操作的意味論がある。

現在、注目されているオブジェクト指向の概念も、抽象データ型の概念の延長線上にあるとみなすこともできる^[1]。オブジェクト指向パラダイムの、多相型 (polymorphic type)、クラス、属性の継承、多重継承は、オブジェクト間で概念や情報の共有を可能にし、プログラムの記述を非常にコンパクトにする。またこれらの概念は、ひとの日常生活における事物の認識方法と共通点が多く、ひとの直観や思考との親和性もよい。また、オブジェクト指向の概念は、いろいろな言語に採用され始めている。た

たとえば Common Lisp では CLOS として、C では C++ として、Prolog では ESP として実現されている。

本稿では抽象データ型に対してオブジェクト指向パラダイムの、多相型、クラス、属性の継承、多重継承を議論できる順序ソート代数に基づく仕様記述法について述べる。この方法は、記述能力が高い反面、仕様の意味を始代数によって簡潔に行うが、その存在が保証されないことも起こる。このため始代数の存在判定算法について述べる。

付録に、順序ソート代数に基づく代数的仕様記述法の形式的側面について補足した。これは、代数的仕様記述法の中心概念であるシグニチャ、代数、準同形写像などの形式的定義を文献^[9]に従って与えたものである。

2. 抽象データ型と代数的仕様記述法

2.1 抽象データ型

抽象データ型の概念は 1960 年代から存在し、現在ではその重要性が広く認識されている。抽象データ型の概念はおよそ次のようにいえる。

抽象データ型は、データ対象とその上の演算群をまとめたものである。この演算によってだけデータ対象を生成、操作できる。したがって演算がデータ対象の外から見える挙動を完全に特徴付ける。それぞれの演算は手続き、あるいは関数として定義する。抽象データ型はその実現を隠し公開しない、つまりデータ対象の表現や演算の具体的な算法を完全に抽象・捨象する。

抽象データ型は仕様と実現とを分離する。このような考えにより、プログラム作成において次のような多くの利点を得られる。

- 1) 正当性証明……正当性証明は、プログラムが仕様を満足することを証明することである。仕様とその実現を分離し、その仕様を代数的仕様記述法などの形式的な仕様記述法で厳密に記述することにより、正当性証明の実施ができるようになる。
- 2) 型の利用……仕様を抽象データ型として記述する場合、仕様記述すべき対象を集合とその上の演算としてモデル化する。そのモデル化の過程は、仕様対象を型の概念で認識し整理する過程である。このように型を意識して仕様記述するため、仕様のレベルで演算の型の不一致の検出が可能である。
- 3) モジュール化……モジュール化は、複雑なシステムをできるだけ独立した部分に分割して、一度に扱う複雑さを減少させようという概念である。抽象データ型はデータとその上の演算群として、これが自然なモジュール化の単位となる。
- 4) 情報隠蔽……情報隠蔽は不必要な情報は公開しないことであり、より積極的に必要最小限の情報だけを提供し、そのほかの情報は見ようとしても見られなくすることである。抽象データ型では、必要最小限の情報は演算間の関係であり、データの表現や演算の算法は隠蔽すべき情報と考える。こうして演算の高速化などのためにデータ構造や演算の算法を変更しても、それらは隠蔽情報の変更であり外部に影響を与えないですむ。

2.2 代数的仕様記述法

代数的仕様記述法では抽象データ型を代数とみなす。代数とは、抽象代数学の用語でいくつかの演算が定義された集合のことであり、集合とその上の演算に注目したものである。代数的仕様記述法では、抽象データ型をデータ上の演算が満たす関係を等式の形で与えて定義する。まず簡単な例としてスタックの仕様を図1に示す。

1	<u>abstractDataType</u>	stack
2	<u>sorts</u>	stack, item, bool
3	<u>signatures</u>	newstack :→ stack
4		push : stack item → stack
5		pop : stack → stack
6		top : stack → item
7		empty? : stack → bool
8	<u>variables</u>	s : stack
9		i : item
10	<u>equations</u>	empty? (newstack) = true
11		empty? (push(s, i)) = false
12		pop (newstack) = newstack
13		pop (push(s, i)) = s
14		top (push(s, i)) = i
15	<u>end</u>	

図1 スタックの仕様
Fig.1 Specification of stack

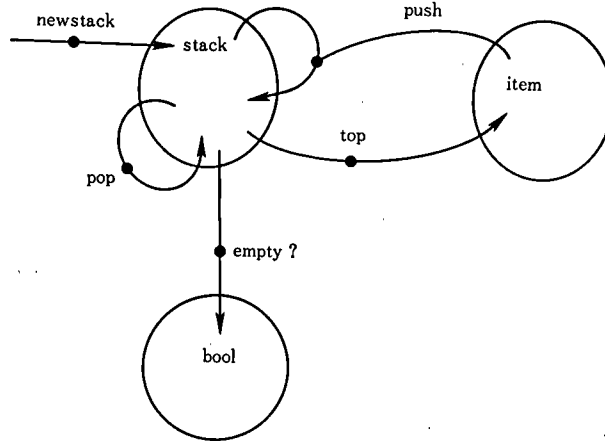


図2 ソートと演算の関係
Fig.2 Relation between sort and operator

この仕様の説明を次に示す。

1行目は、定義する抽象データ型の名前を記述する。

2行目は、この定義に出現する型の名前を記述する。この名前をソートという。ソートは名前であるが混乱の恐れがない時は、記述を簡潔にするために名前が指す集合そのものもソートと呼ぶ。

3行目から7行目は、シグニチャ、つまりこの抽象データ型を規定するそれぞれの演算子の名前、引数の型の並び、演算の結果の型を記述している。これを図2に示す。

8, 9 行目は, 次の等式で参照する変数名とその型を記述している。

10 行目から 14 行目は, 演算が満たす関係を等式の形で記述する。

演算は次のようなものと考えている。

newstack: 空のスタックを値として返す。

push: 新しいアイテムをスタックの最上位に押し込み, その状態のスタックを返す。

pop: スタックの最上位にあるアイテムを取り除いたスタックを返す。

top: スタックの最上位にあるアイテムを値として返す。

empty?: スタックが空かどうかテストし, その結果を論理値で返す。

2.3 仕様とその意味

スタックの仕様記述が何を意味するかを厳密に考えよう。そのために, 記述した仕様から形式的なモデルを生成する。モデルの生成は仕様から形式的に集合と演算を生成することにより行う。

状況を整理する。ソート集合 S として $\{\text{stack}, \text{item}, \text{bool}\}$ がある。ソートとは型の名前である。シグニチャとして,

newstack $\in \Sigma \epsilon, \text{stack}$

push $\in \Sigma \text{stack}, \text{item}, \text{stack}$

pop $\in \Sigma \text{stack}, \text{stack}$

top $\in \Sigma \text{stack}, \text{item}$

empty? $\in \Sigma \text{stack}, \text{bool}$

がある。シグニチャは, ソートの有限列とソートの組を添字とする集合族で, その集合の要素は演算名である。これに演算間の関係が等式として与えられている。実はソート $s \in S$ に対してキャリアと呼ぶある集合 A_s が付随して存在し, その上にシグニチャの演算名 $\sigma \in \Sigma$ に対し, 対応して A_s たちの上の演算 $A\sigma$ が存在する。この演算が等式を満たすと考える。このキャリア A_s と, その上の演算 $A\sigma$ を数学の用語に従って代数 (ソートが多数存在するので多ソート代数) と呼び, これを抽象データ型と考えるのである。

記述された仕様から形式的に, 具体的なキャリア A_s とその上の演算 $A\sigma$ をつくることができる。

まず項 (term) を考える。項とはシグニチャに出現した演算から次のようにして構成した式のことである。

1) すべてのソートの定数は, そのソートの項である。

2) 演算 $\sigma \in \Sigma s_1, s_2, \dots, s_n, s_k$ とソート s_1, s_2, \dots, s_n の項 t_1, t_2, \dots, t_n に対して, $\sigma(t_1, t_2, \dots, t_n)$ は, ソート s_k の項である。

たとえば, newstack はソート stack の項であり, i, i_1, \dots, i_k をソート item の項とすると,

push(newstack, i)

push(... push(push(newstack, i 1), i 2) ..., i k)

push(... pop(...) ..., i)

などはソート stack の項である。この項の集合をキャリア A_s とする。たとえば, 上記

のような項は $stack \in S$ に対するキャリア A_{stack} の要素である。

演算 $A\sigma$ は項を引数とし、項を結果として返す演算として次のように定義する。引数の型を $\omega = s_1, \dots, s_n$, 結果の型を s とする演算名 σ の演算 $A\sigma: A\omega \rightarrow As$ を、ソート s_1, s_2, \dots, s_n の項 t_1, t_2, \dots, t_n に対して、項 $\sigma(t_1, t_2, \dots, t_n)$ を結果の値として返す演算とする。

これで、記述された仕様から形式的な集合 As と演算 $A\sigma$ が具体的に生成できた。仕様から生成した As と $A\sigma$ の全体を考え、 T_{stack} と書いて、自由項代数 (free term algebra) という。等式を持たないとき、仕様が意味するものは、この自由項代数である。

つぎに等式を考えよう。等式は、項 t, t_1, t_2 に対して、

- 1) $t = t$
- 2) $t = t_1 \quad \longrightarrow \quad t_1 = t$
- 3) $t = t_1, t_1 = t_2 \quad \longrightarrow \quad t = t_2$

という同値関係を自由項代数の上に定義するものと考え。たとえば、等式を項の上に適用すれば、

```

push(push(newstack, i1), i2)
  = pop(push(push(push(newstack, i1), i2), i3))
  = pop(pop(push(push(push(push(newstack, i1), i2), i3), i4)))
  = ...

newstack
  = pop(push(newstack, i))
  = pop(pop(push(push(newstack, i1), i2)))
  = ...
    
```

などが得られる。項 t に対して同値な項全体を t の同値類 (equivalence class) という。同値類は同値関係から空ではなく、異なる同値類の間では共通項を持たない。同値類の全体、つまり自由項代数の等式による類別 (classification) を商代数 (quotient algebra) という。この商代数がスタックの仕様から形式的に導いたモデルであり、このモデルを仕様の意味とする。

この商代数は、仕様を満たす代数の中で始代数と呼び、いちばん詳しい構造を持った代数になっている。言い換えると、この仕様を満たすどんな代数を持ってきても、商代数からその代数に対して準同形写像を定義できる。準同形写像とは、集合と演算の構造を保存する代数から代数への写像である。構造を保存するとは、写像前の代数で成立していた演算と集合の関係を写像後の代数でも成立させることである。

また、二つの代数が互いに準同形写像の関係にあるとき、すなわち両者の構造がまったく等しいとき、その二つの代数は同形である、または同形写像の関係にあるという。

ここで仕様の実現とは何かを考えてみると、この始代数と同形な関係にある任意の代数のことであるといえる。たとえば、スタックを配列やリストで実現しても、それらを代数として見ると、始代数と同形写像の関係にある。最終的に仕様の意味するモデルは、仕様を満たす代数の始代数に同形な代数のクラスであるといえる。

3. 順序ソート代数への拡張

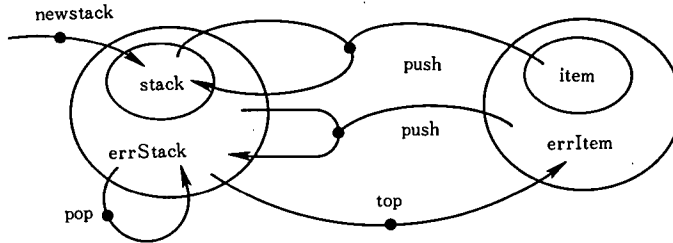
オブジェクト指向パラダイムの、多相型、クラス、属性の継承、多重継承は、オブジェクト間で概念や情報の共有を可能にし、プログラムの記述を非常にコンパクトにする。また、これらの概念はひとの日常生活における事物の認識方法と共通点が多く、ひとの直観や思考との親和性もよい。オブジェクト指向のこれらの有用な概念を代数的仕様記述法に導入するために、抽象データ型のモデルを従来の多ソート代数から、順序ソート代数に拡張する。

3.1 順序ソート代数に基づく代数的仕様記述法

仕様記述の例として、さきほどのスタックにエラー処理を加えたものを図3に示す。新しいソート `errStack`, `errItem` は、ソート `stack` と `item` にエラー状態を追加したソートである。3, 4行目でこれらのソート間に包含関係を定義している。

1	<u>abstractDataType</u>	stack
2	<u>sorts</u>	stack, item, bool, errStack, errItem
3	<u>sortOrder</u>	stack \leq errStack
4		item \leq errItem
5	<u>signatures</u>	newstack : \rightarrow stack
6		push : stack item \rightarrow stack
7		push : errStack errItem \rightarrow errStack
8		pop : errStack \rightarrow errStack
9		top : errStack \rightarrow errItem
10		serr : \rightarrow errStack
11		ierr : \rightarrow errItem
12	<u>variables</u>	s : stack
13		i : item
14		es : errStack
15		ei : errItem
16	<u>equations</u>	pop(push(s, i)) = s
17		top(push(s, i)) = i
18		push(es, ierr) = push(serr, ei) = serr
19		top(newstack) = top(serr) = ierr
20		pop(newstack) = pop(serr) = serr
21	<u>end</u>	

(a)仕様記述



(b)ソートと演算の関係

図3 スタックの順序ソート代数に基づく仕様

Fig. 3 Specification of stack based on order-sorted algebra

3.2 仕様とその意味

多ソート代数と同様に、順序ソート代数に基づく仕様記述の意味を考える。

ソート集合 S として、 $\{ \text{stack}, \text{errStack}, \text{item}, \text{errItem} \}$

ソートの順序として、 $\text{stack} \leq \text{errStack}, \text{item} \leq \text{errItem}$

シグニチャとして

$\text{newstack} \in \Sigma_{\epsilon}, \text{stack}$

$\text{push} \in \Sigma \text{ stack}, \text{item}, \text{stack}$

$\text{push} \in \Sigma \text{ errStack}, \text{errItem}, \text{errStack}$

$\text{pop} \in \Sigma \text{ errStack}, \text{errStack}$

$\text{top} \in \Sigma \text{ errStack}, \text{errItem}$

$\text{serr} \in \Sigma_{\epsilon}, \text{errStack}$

$\text{ierr} \in \Sigma_{\epsilon}, \text{errItem}$

ソートに順序を導入したことにより、順序ソート代数では次のような制約を集合と演算の間に設ける。

- 1) s, s' をソートとし、 A_s と $A_{s'}$ をキャリアとするととき、
 $s \leq s'$ ならば A_s は $A_{s'}$ の部分集合である。
- 2) w_1, w_2 をソートの有限系列とし、 s_1, s_2 をソートとするととき、
 ある演算 σ があって、 $\sigma \in \Sigma w_1, s_1, \sigma \in \Sigma w_2, s_2, w_1 \leq w_2, s_1 \leq s_2$ ならば、キャリア A_{w_1} 上では、
 $A_{\sigma}: A_{w_1} \rightarrow A_{s_1}$ と $A_{\sigma}: A_{w_2} \rightarrow A_{s_2}$ は等しい。

したがって、自由項代数においてもソートの順序関係に従い、項集合上に包含関係を制約として加える。たとえば、

newstack
 $\text{push}(\text{newstack}, i)$
 $\text{push}(\dots \text{push}(\text{newstack}, i) \dots, \text{in})$
 $\text{pop}(\text{push}(\dots \text{push}(\text{newstack}, i) \dots, \text{in}))$

などはソート push の項であり、ソート errStack の項でもある。またエラー状態を示す次の項は、ソート errStack だけの項である。

$\text{pop}(\text{newstack})$
 $\text{pop}(\text{push}(\dots \text{pop}(\text{newstack}) \dots, \text{in}))$
 serr
 $\text{push}(\text{serr}, i)$

このように stack のキャリアは errStack のキャリアの部分集合となっている。

つぎに同値関係による同値類を考えよう。多ソートならば異なるソートが指すキャリアの要素は同値な要素を持たない。しかし、順序ソートではキャリアの包含関係があるため、異なるソートにまたがる同値類を考える必要がある。この異なるソートにまたがる同値類による商代数を考えるために、後で述べる接続性という条件を仮定する。この条件は、すべてのソートにそれを含む最大ソートが必ず一つだけ存在することを要請する。各ソートについて同値類を考えるとき、そのソートの最大ソートで同値類を作成することにより、順序関係があるソートにまたがって同値類を作成できる。

こうして順序ソート代数の商代数を作成する。

このようにして仕様から形式的に導いた商代数を、この仕様の意味を与えるモデルと考えたい。しかし、この商代数の存在も仕様に接続性を仮定して初めて考えられたように、仕様に後で述べる正則性を仮定しないと、この商代数は始代数にならない。すなわち、仕様を満たす代数の中に、この商代数より詳しい構造を持った代数が存在する。したがって記述された仕様が接続性と正則性を満たさない時は、仕様から始代数モデルを形式的に作成できず、その仕様に宣言の意味を与えることができない。接続性と正則性を仕様を満たす時は、仕様の意味するモデルは仕様を満たす代数の始代数に同形な代数のクラスである。

3.3 順序ソート代数の特徴

順序ソートに基づく順序ソート代数 (Order-Sorted Algebra) は、多ソートでは扱えなかった集合間の包含関係、部分演算、演算の多重継承、多義演算が表せ記述能力が高い。これらについて次に述べる。

3.3.1 集合間の包含関係

仕様記述すべき対象が持つ集合の間の包含関係を扱える。これはソート間に集合の包含関係による半順序関係 \leq を付けることにより記述する。たとえばスタックでは正常な状態の集合 `stack` と、それにエラー状態を含ませた集合 `errStack` を考えた。そしてこの関係を `stack \leq errStack` のように記述した。

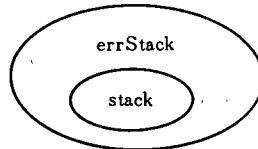


図4 集合の包含関係

Fig. 4 Inclusion relation of sets

3.3.2 部分演算

演算 f の定義域がソート s で示す集合の部分集合に対応するとき、演算 f はソート s の部分演算 (partial operator) であるという。部分演算を使えば、ある集合の部分集合に対してのみ演算を定義できる。スタックの例では、二つの `push` 演算のうちの一つが `errStack` と `errItem` に対して部分演算となっている。

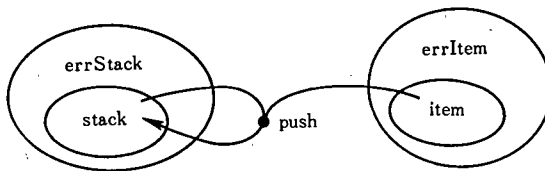


図5 部分演算

Fig. 5 Partial operator

3.3.3 演算の多重継承

ソートの包含関係が $X \leq A$ の場合に、 A 上で定義された演算 f は、その下位ソート X から参照可能である。たとえば、関数の定義が $f: A \rightarrow A$ ならば、演算 f の定義域 A をその下位ソート X で置き換えた、 $f: X \rightarrow A$ が成立する。このように、上位ソートで定義した演算を下位ソートに引き継がせることを演算の継承 (inheritance) という。また $X \leq A, X \leq B$ の場合、 X は上位ソートの A と B から演算を引き継ぐ。このように複数の上位ソートから各々の演算を引き継ぐことを多重継承 (multiple inheritance) という。スタックの例では、 $\text{pop}: \text{errStack} \rightarrow \text{errStack}$ であるので、定義域の下位ソートである stack に演算が継承され $\text{pop}: \text{stack} \rightarrow \text{errStack}$ が成立する。

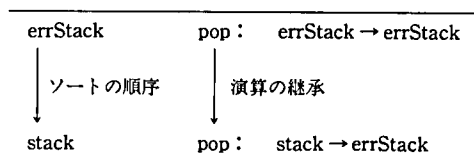


図6 演算の継承

Fig.6 Inheritance of operator

3.3.4 多義演算

一つの演算記号を複数の演算に対して用いるとき、その演算記号を多義演算記号 (overloaded operator symbol) という。たとえば、スタックの演算記号 push はスタックにデータを追加する演算と、エラー状態を引き継ぐ演算の二つの演算に対する記号として使われている多義演算記号である。

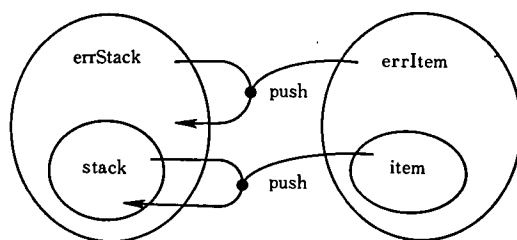


図7 多義演算

Fig.7 Overloaded operator

4. 始代数の存在とその判定

代数的仕様記述法の形式的意味論では、記述した仕様の意味を始代数で定める。そして、意味を定める始代数と同形写像の関係にある任意な代数の実現を仕様の実現と考える。しかし、順序ソート代数に基づく方法では記述した仕様から形式的に得られる自由項代数の商代数が、仕様を満たす代数のクラスの中の始代数になることが保証されない。始代数であることを保証するための十分条件は、付録で述べる正則性 (定義9)、連続性 (定義12) である (命題3)。

4.1 正則性

正則性とは、演算の多重継承が競合しないことである。演算が競合しないとは、同名の演算の多重継承があったとき、ソートの順序関係によりいちばん近くにある演算の定義を一意に決定できることである。形式的にいうと次のようになる。

ソート集合 S の元の有限系列からなる集合を S^* とするとき、

$\omega_1, \omega_0 \in S^*, \omega_0 \leq \omega_1, \sigma \in \Sigma \omega_1, s_1$ ならば、

$\omega_0 \leq \omega, \sigma \in \omega, s$ である ω と s の組 $\langle \omega, s \rangle$ の中で、ソートの順序関係により順序が最低の組が存在することである。

4.2 接続性

接続性とは、すべてのソート $s \in S$ に対し、最大ソート $m(s)$ が存在することである。形式的にいうと次のようになる。

すべての $s \in S$ に対して $m(s)$ が存在して、

$s \leq m(s)$ かつ

$s \leq s_1$ であるすべての s_1 について $s_1 \leq m(s)$ となる。

4.3 始代数の存在しない例

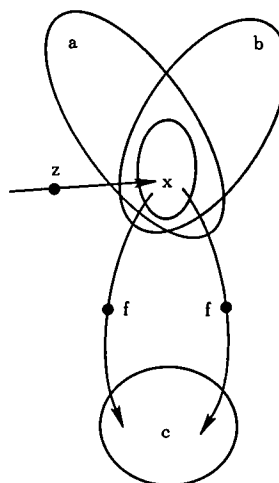
接続性と正則性がなく、仕様の意味を定める始代数が存在しない例を示す。ソートの集合を $\{x, a, b, c\}$ とし、その間の包含関係を $x \leq a, x \leq b$ とする。ソート x の示す集合を $\{z\}$ とする。同一の演算記号 f を持つ相異なる演算が、次のように定義されていたとする。 $f: a \rightarrow c, f: b \rightarrow c$ このとき、ソート c の中の値である $f(z)$ に対応する意味は、 z にどちらの関数を適応したかで2通り存在し一意に決定できない。この例の仕様記述を図8に示す。

この仕様から生成した自由項代数の各ソートの要素を図9(a)に示す。またこの仕

```

abstractDataType  notInit
  sorts           x, a, b, c
  sortOrder      x ≤ a
                 x ≤ b
  signatures     f : a → c
                 f : b → c
                 z : → x
end
    
```

(a) 仕様



(b) ソートと演算の関係

図8 始代数の存在しないソート演算の関係

Fig. 8 Relation between sort and operator, which has no initial algebra

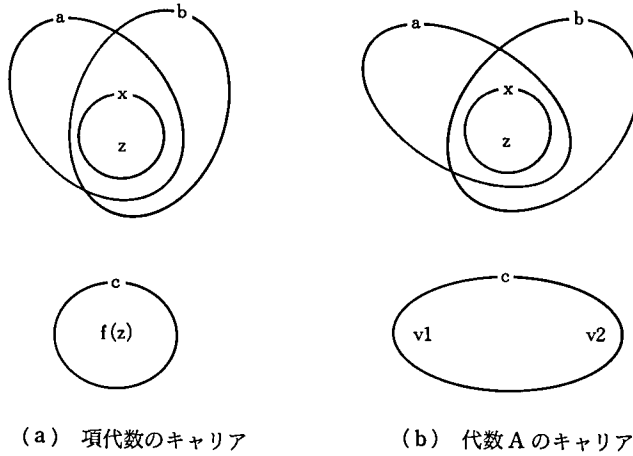


図9 項代数とより詳しい構造を持つ代数 A

Fig.9 Term algebra and algebra A which has finer structure than it

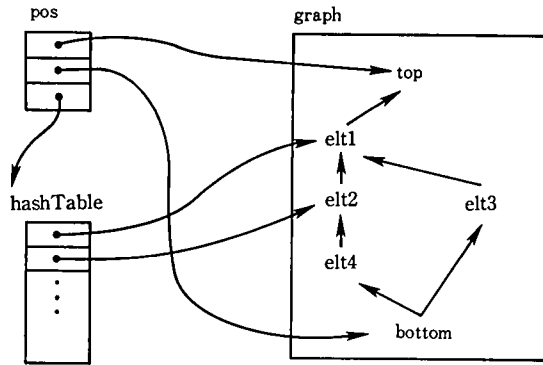


図10 POS マネージャのデータ構造

Fig.10 Data structure of POS manager

様を満たす代数として、図9(b)のような要素を持つ代数 A が存在する。この両者を比べると、代数 A の方が詳しい構造を持っている。したがって、仕様から形式的に作成した自由項代数は、仕様を満たす代数のクラスの中で始代数にはなっていない。このケースでは、演算記号 f の多重継承が自由項代数が始代数にならない原因となっている。

4.4 始代数判定アルゴリズム

始代数であることを保証するための十分条件は、付録の正則性(定義9)、接続性(定義12)である(命題3)。記述した仕様の自由項代数の商代数が始代数かどうかを判定するために、この条件を判定する演算を作成した。

4.4.1 半順序関係

ソート集合 S の半順序関係 \leq 、およびソートの有限系列上の半順序関係を扱うために、POS (Partially Ordered Set) マネージャを作成した。これは抽象化した半順序関係を扱うもので、半順序関係を持った要素をノード、半順序関係をアークとして全

体を有向グラフとして扱う。また各ノードに直接アクセスするためにノード名を指標、ノードを値とするハッシュ・テーブルを持たせた (図 10)。

4.4.2 接続性, 正則性の判定アルゴリズム

接続性, 正則性を判定するアルゴリズムの概略をアルゴル・ライクの言語を使用して図 11, 図 12 に示す。ここでは演算の引数の型の並びをアリティ, 演算の結果の型をコアリティと呼んでいる。

```

begin
  sortPosをソートの順序関係より構成したPOSとする;
  topSet ← sortPosの中の上方に要素が存在しない要素の集合;
  bottomSet ← sortPosの中の下方に要素が存在しない要素の集合;
  each bottom ∈ bottomSet do
    begin
      upperSet ← 要素bottomの上方の要素からなる集合;
      if topSet ∩ (bottom ∪ upperSet)の要素の個数 ≠ 1 then
        接続性は満たされていない;
      end
    end
  接続性は満たされている;
end

```

図11 接続性 判定アルゴリズム
Fig.11 Decision algorithm for 'cohereny'

5. あとがき

順序ソート代数に基づく代数的仕様記述法は, 集合間の包含関係, 部分演算, 演算の多重継承, 多義演算を扱え記述能力が高いが, 記述した仕様の意味を定める始代数の存在が保証されないことがある。そこで, 本研究では記述した仕様の始代数の存在を確認するため, 始代数の存在判定手続きを実現した。

最後に, この研究の機会を与えて戴いた電子技術総合研究所の棟上昭男部長, 日本ユニシスの米口肇部長, 渡部義維部長ならびに情報処理学会の記号処理研究会に提出した論文の共著者である, 電子技術総合研究所の二木厚吉室長, 戸村哲技官, 株式会社 SRA の中川中氏に感謝の意を表す。


```

begin
  allOperator ← 仕様より定義された演算 (演算記号, アリティ, コアリティの組) の集合;
  each  $\sigma \in$  (allOperatorの各要素が持つ演算記号を要素とする演算記号の集合)
  do
    begin
      preOverLoadSet ← allOperatorの中で演算記号  $\sigma$  を持つ演算の集合;
      if preOverLoadSetの要素数が2以上 then
        each  $i \in$  preOverLoadSetの各演算が持つ引数の個数を要素とする整数の集合;
        begin
          overLoadSet ← preOverLoadSetの中で引数の個数が  $i$  である演算の集合;
          if overLoadSetの要素数が2以上 then
            begin
              posを空集合とする;
              each  $\text{arity} \in$  overLoadSetの各要素が持つアリティを要素とする集合;
              addLowerArity(pos, arity)
              each  $\text{elt} \in$  posに含まれる要素の集合 do
                if not(leastOperator?(pos, elt, overLoadSet)) then
                  正則性は満たされていない;
            end
          end
        end
      end
      end
    end
  end
  正則性は満たされている;
end
procedure addLowerArity(pos, arity)
begin
  ソートの直積集合 (演算の定義域) 上の半順序集合posに要素arityが存在しなければ, それを追加する;
  each  $\text{elt} \in$  要素arityの直下の要素の集合 do
    if 要素eltがposの要素でない then
      addLowerArity(pos, elt);
  end
end
procedure leastOperator? (pos, elt, operatorSet)
begin
  arityEltSet ← 集合operatorSetに含まれる演算の定義域に対応する, ソートの直積集合上の半順序集合posの要素の集合;
  upperArityEltSet ←  $\text{arityEltSet} \cap$  (半順序集合posの要素eltとその上方の要素からなる集合);
  upperCoarityEltSet ← 半順序集合upperArityEltSetの要素をアリティとするoperatorSet中の演算のコアリティの集合;
  leastAritySet ← 半順序集合upperArityEltSetの中の要素のうち, その中の他の要素を下方に持たない要素の集合;
  leastCoaritySet ← 半順序集合upperCoarityEltSetの中の要素のうち, その中の他の要素を下方に持たない要素の集合;
  return (and (leastAritySetの要素数は1)
              (leastCoaritySetの要素数は1)
              (アリティがleastAritySetの要素であり, コアリティがleastCoaritySetの要素である演算がoverLoadSetに存在する))
end

```

図12 正則性 判定アルゴリズム
Fig.12 Decision algorithm for 'regularity'

付録 順序ソート代数の形式的定義

多ソート代数を拡張した順序ソート代数について、代数的仕様記述法の中心的概念であるシグニチャ、代数、準同形写像などの定義を文献^[5]に従って与える。

定義1 シグニチャ (signature)

ソートの集合を S とし、 S 上の半順序関係を \leq とし、 Σ を演算記号を要素とする集合の族で $\Sigma = \{\Sigma_{w,s} | w \in S^*, s \in S\}$ とする。ここで S^* は、 S の元からなる有限系列の集合である。 w, s はそれぞれ演算の引数と結果のソートを示し、 w をアリティ (arity)、 s をコアリティ (coarity) と呼ぶ。 S と \leq と Σ の組を順序ソート・シグニチャと呼び、 (S, \leq, Σ) で表す。混乱のおそれがない時は、 (S, \leq, Σ) を Σ で表す。

\leq を S^* 上の長さの等しい系列 (string) に次のように拡張する。

$$(s_1..s_n \leq s'_1..s'_n) \leftrightarrow (s_i \leq s'_i \text{ for } i=1..n)$$

S^* , S 上の組 $\langle w, s \rangle$ についても \leq を次のように拡張する。

$$\langle w, s \rangle \leq \langle w', s' \rangle \leftrightarrow (w \leq w' \text{ and } s \leq s')$$

定義2 代数 (algebra)

(S, \leq, Σ) を順序ソート・シグニチャとする。順序ソート代数 A は、 (S, \leq, Σ) 代数 A と表し、次にあげる二つの条件を満たすように、各ソート $s \in S$ に台 (carrier) と呼ぶ集合 A_s を割り当て、 Σ の各演算記号 $\sigma \in \Sigma_{w,s}$ に具体的な演算 $A\sigma: A_w \rightarrow A_s$ を割り当てることにより得られる。

- ① $(s \leq s' \text{ in } S) \rightarrow A_s \subseteq A_{s'}$
- ② $\{\sigma \in (\Sigma_{w_1, s_1} \cap \Sigma_{w_2, s_2}), s_1 \leq s_2, w_1 \leq w_2\}$
 $\rightarrow \{(A\sigma: A_{w_1} \rightarrow A_{s_1}) = (A\sigma: A_{w_2} \rightarrow A_{s_2}) \text{ on } A_{w_1}\}$

定義3 準同形写像 (homomorphism)

(S, \leq, Σ) を順序ソート・シグニチャとし、代数 A と代数 B を (S, \leq, Σ) 代数とする。順序ソート準同形写像 h は、 (S, \leq, Σ) 準同形写像 $h: A \rightarrow B$ と表し、ソート集合 S を添字とする関数族 $h = \{h_s: A_s \rightarrow B_s | s \in S\}$ であり次の条件を満たすものである。

- ① $h_s(A\sigma(a)) = B\sigma(h_w(a))$ for each $\sigma \in \Sigma_{w,s}$ and $a \in A_w, w \neq \varepsilon$
- ② $h_s(A\sigma) = B\sigma$ for $w = \varepsilon$
- ③ $(s \leq s' \text{ and } a \in A_s) \rightarrow h_s(a) = h_{s'}(a)$

定義4 同型写像 (isomorphism)

次の条件を満たす準同形写像 $h: A \rightarrow B$ を同型写像という。

$$\exists (h^{-1}: B \rightarrow A)(h^{-1} \circ h = 1_A, h \circ h^{-1} = 1_B)$$

定義5 項代数 (term algebra)

(S, \leq, Σ) を順序ソート・シグニチャとする。順序ソート項代数 $\tau[\Sigma]$ は、各ソート $s \in S$ に対する台として次の条件を満たす最小の集合 $\tau[\Sigma]_s$ を割り当てる。

- ① $\Sigma \varepsilon, s \subseteq \tau[\Sigma]_s$
- ② $s' \leq s \rightarrow \tau[\Sigma]_{s'} \subseteq \tau[\Sigma]_s$
- ③ $\sigma \in \Sigma_{w,s}, w = s_1..s_n, t_i \in \tau[\Sigma]_{s_i} \text{ for } i=1, \dots, n$
 $\rightarrow \sigma(t_1 \dots t_n) \in \tau[\Sigma]_s$

各演算 $\sigma \in \Sigma_{w,s}$ $\tau[\Sigma]_w \rightarrow \tau[\Sigma]_s$ には引数が $t_1..t_n$ ならば、系列 $\sigma(t_1 \dots t_n)$ を結果として返す演算を割り当てる。

定義 6 自由項代数 (free term algebra)

(S, \leq, Σ) を順序ソート・シグニチャとする。変数集合族 $X = \{X_s | s \in S\}$ は、次の性質を持つとする。

- ① $X_s \cap \Sigma w, s = \emptyset$
- ② $s_i \neq s_j \rightarrow X_{s_i} \cap X_{s_j} = \emptyset$

このとき順序ソート・シグニチャ $(S, \leq, \Sigma(X))$ を次のように定義する。

- ① $\Sigma(X)_{\epsilon, s} = \Sigma_{\epsilon, s} \cup X_s$
- ② $\Sigma(X)_{w, s} = \Sigma_{w, s}$ for $w \neq \epsilon$

順序ソート・シグニチャ $(S, \leq, \Sigma(X))$ から定まる。順序ソート項代数 $\tau[\Sigma(X)]$ を X 上の順序ソート自由項代数という。

定義 7 始代数 (initial algebra)

Σ を順序ソート・シグニチャとする。 Σ 代数 I がすべての Σ 代数のクラスの中で始代数であるとは、任意の Σ 代数へ I からの順序ソート準同形写像が一意的に存在することである。

定義 8 割当 (assignment)

A を順序ソート Σ 代数とし、 X を変数集合族とする。 $f: X \rightarrow A$ はソート集合 S を添字とする関数族であり X から A への割当と呼ぶ。 f を自然に拡張した $f^*: \tau[\Sigma(X)] \rightarrow A$ は、順序ソート準同型写像になる。

定義 9 正則 (regularity)

順序ソート・シグニチャ (S, \leq, Σ) が正則であるとは、

$$w_1, w_0 \in S^*, w_0 \leq w_1, \sigma \in \Sigma w_1, s_1 \rightarrow \exists \langle w, s \rangle (\sigma \in \Sigma w, s, w_0 \leq w, \text{least}(\langle w, s \rangle))$$

が成立することである。

正則であるとは、ソートの有限系列の各要素に対して演算の継承が競合しないことを意味する。演算が競合しないとは、ソートのある有限系列 w_0 で、同一演算記号に複数の演算の継承が存在した場合、ソートの順序関係により、 w_0 以上で最小のソートを持つ演算の定義を一意的に見つけられることを意味する。

命題 1 最小ソート (least sort)

Σ が正則ならば、各 $t \in \tau[\Sigma]$ は、 $t \in \tau[\Sigma]_s$ を満たす最小ソート $s \in S$ を持つ。 t の最小ソートを $LS(t)$ と表す。

定義 10 等式 (equation)

(S, \leq, Σ) を正則な順序ソート・シグニチャとする。 Σ 等式は、 $\langle X, t, t' \rangle$ である。ここで X は変数集合族、 t, t' は $\tau[\Sigma(X)]$ の要素で、 t と t' の最小ソートである $LS(t)$ と $LS(t')$ の間には (S, \leq) による包含関係が存在しなければならない。

Σ 代数 A が Σ 等式 $\langle X, t, t' \rangle$ を満たすことを $(\forall X)t=t'$ on A のように表し、次のように定義する。

$$(\forall X)t=t' \text{ on } A \iff \text{すべての割当 } f: X \rightarrow A \text{ について } f^*_{LS(t)}(t) = f^*_{LS(t')}(t')$$

同様に Σ 代数 A と Σ 等式の集合 Γ に対して、 Γ のすべての等式が A 上で成り立つとき、かつその時に限り A は Γ を満たすと定義する。

定義 11 合同関係 (congruence)

(S, \leq, Σ) を順序ソート・シグニチャとし, A を順序ソート Σ 代数とする. 次の条件を満たすとき, その関係を A 上の順序ソート Σ 合同関係 $\equiv_s = \{ \equiv_s | s \in S \}$ という.

- ① 各ソート $s \in S$ に対して \equiv_s は A_s 上の同値関係である.
- ② 各 $\sigma \in \Sigma w, s (w = s_1 \cdots s_n)$ で $i = 1, \dots, n$ について
 $a_i \equiv_s a'_i, a_i, a'_i \in A_{s_i}$ ならば $A\sigma(a_1, \dots, a_n) \equiv_s A\sigma(a'_1, \dots, a'_n)$
- ③ 各 $s, s' \in S$ で $s \leq s', a, a' \in s$ ならば $a \equiv_s a' \leftrightarrow a \equiv_{s'} a'$

任意の $a \in A_s$ の, \equiv_s による同値類 (equivalence class) を $[a]$ で表し, 次のように定義する.

$$[a] = \{ x | a \equiv_s x, x \in A_s \}$$

定義 12 連接 (coherency)

半順序集合 (S, \leq) が連接であるとは, すべてのソート $s (s \in S)$ に対し $\max(s)$ が存在し,

- ① $s \leq \max(s)$
- ② $s \leq s_1$ ならば $s_1 \leq \max(s)$

が成立することである. これは各ソートに対して最大ソートが一意に決まることである.

定義 13 商代数 (quotient)

(S, \leq, Σ) を連接な順序ソート・シグニチャとし, A を順序ソート Σ 代数, \equiv を A 上の順序ソート合同関係とする. \equiv による A の商代数を順序ソート Σ 代数 A/\equiv と表し, 次のように定義する.

各最大ソートの台を

$(A/\equiv)_{\max(s)} = A_{\max(s)} / \equiv_{\max(s)}$ と定義し, 各 $s \in S$ について, その台 $(A/\equiv)_s$ を $\{ q_{\max(s)}(a) | a \in A_s \}$ とする. ただし $q_{\max(s)}$ は

$$q_{\max(s)}: A_{\max(s)} \rightarrow (A/\equiv)_{\max(s)}$$

で, 引数 $a \in A_{\max(s)}$ に $[a]$ を結果として返す関数とする.

各演算 $\sigma \in \Sigma w, s$ は次のように定義する.

$$(A/\equiv)\sigma([a_1], \dots, [a_n]) = [A\sigma(a_1, \dots, a_n)]$$

定義 14 推論規則 (deduction rule)

Σ を正則で連接な順序ソート・シグニチャとする. 等号論理における推論規則を次のように定義する.

- ① 反射則

$$(\forall X)t = t$$

- ② 対称則

$$(\forall X)t = t' \rightarrow (\forall X)t' = t$$

- ③ 推移則

$$(\forall X)t = t', (\forall X)t' = t'' \rightarrow (\forall X)t = t''$$

- ④ 合同則

$t \in \tau[\Sigma(X)], f^*, f^{**}: \tau[\Sigma(X)] \rightarrow \tau[\Sigma(Y)], x \in X$ とすると

$$(\forall Y)f^*(x) = f^{**}(x) \rightarrow (\forall Y)f^*(t) = f^{**}(t)$$

⑤ 代入則

$$\begin{aligned}
 & (\forall X)t=t' \\
 & \rightarrow (\forall Y)f^*(t)=f^*(t')
 \end{aligned}$$

定義 15 等式と合同関係

Σ を正則で連接な順序ソート・シグニチャとし, Γ を Σ 等式の集合とする。順序ソート合同関係 $\approx \Gamma(X)$ を $t, t' \in \tau[\Sigma(X)]$ として, 次のように定義する。

$$\begin{aligned}
 & t \approx \Gamma(X)t' \leftrightarrow \\
 & (\forall X)t=t' \text{ が } \Gamma \text{ と推論規則から導出可能。}
 \end{aligned}$$

$\approx \Gamma(X)$ による $\tau[\Sigma(X)]$ の商代数として, 定義される順序ソート代数を $\tau[\Sigma(X), \Gamma]$ と表す。

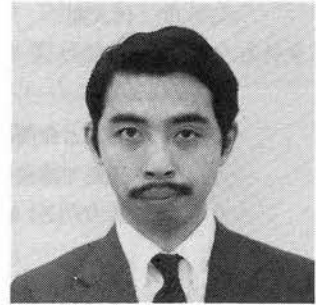
命題 3 始代数

Σ を正則で連接な順序ソート・シグニチャとし, Γ を Σ 等式の集合とする。
 $\tau[\Sigma(\varepsilon), \Gamma]$ は, Γ を満たすすべての順序ソート Σ 代数のクラスにおいて始代数である。

参考文献 [1] K. Futatsugi, J. Goguen, J. Jouannaud, and J. Meseguer, "Principles of OBJ 2", Proc. of 12th ACM Symposium. on Principles of Programming Languages, 1985.
 [2] K. Futatsugi, "An overview of OBJ2", Proc. of France-Japan A. I. and C. S. Symposium 86, Institute for New Generation Computer Technology, Japan, 1986.
 [3] K. Futatsugi, J. Goguen, J. Meseguer, K. Okada, "Parameterized Programming in OBJ 2", Proc. of 9th Int. Conf. on S. E., 1987.
 [4] J. Goguen, C. Kirchner, J. Meseguer, "Concurrent Term Rewriting as a Model of Computation", Proc. Graph Reduction Workshop, Santa Fe, NM, SRI International, Computer Science Lab., Sept. 1986.
 [5] J. Goguen, J. Meseguer, "Order-Sorted Algebra I: Partial and Overloaded Operators, Errors and Inheritance", Technical Report to appear, SRI International, Computer Science Lab., 1987.
 [6] J. Meseguer, J. Goguen, "Initiality, Induction and Computability", Algebraic Methods in Semantics, Cambridge University Press, 1984.
 [7] A. Nakagawa, K. Futatsugi, S. Tomura, T. Shimizu, "Algebraic Specification of Macintosh's QuickDraw Using OBJ2", 10th International Conference on Software Engineering, 1988.
 [8] 有沢, ソフトウェア工学, 岩波書店, 1988.
 [9] 稲垣・坂部, 抽象データタイプの代数的仕様記述法の基礎, (1)多ソート代数と等式論理, 情報処理, Vol. 25, No. 1, (2)抽象データタイプ, 情報処理, Vol. 25 No. 5, (3)抽象型構成子のモデルと始代数意味論, 情報処理, Vol. 25. No. 7, (4)終代数意味論に基づく抽象型構成子の仕様記述, 実現ならびにその検証, 情報処理, Vol. 25. No. 9.
 [10] 木村・米澤, 算法表現論, 岩波書店, 1984.
 [11] 芝山, オブジェクト指向, bit, Vol. 20, No. 6, 1988.
 [12] 清水・戸村・二木・中川, 順序ソート代数仕様と始代数存在判定手続き, 情報処理学会, 記号処理研究会資料, 44-4, 1988.
 [13] 鳥居・二木・真野, プログラミング方法論の展望, 情報処理, Vol. 20, No. 1. 1979.
 [14] 最新 Ada 基準文法書, bit 別冊 1984.
 [15] 二木・中川, 抽象データ型と OBJ 2, bit, Vol. 20, No. 9, 1988.
 [16] J.ロイド, 論理プログラミングの基礎, 産業図書 1987.

執筆者紹介 清水敏彦 (Toshihiko Shimizu)

昭和33年生。56年職業訓練大学校電子科卒業，同年日本ユニシス(株)入社，デジタル回路の解析，開発支援システムの開発を経て，構造解析関連ソフトウェアの保守に従事。現在CAEシステム部所属。情報処理学会，日本ソフトウェア科学会会員。



編集者注 本稿は，情報処理学会，記号処理研究会（1988年1月22日）に発表した「順序ソート代数仕様と始代数判定手続き」（清水敏彦，戸村哲，二木厚吉，中川中）をもとに作成された。

H-CAD とプレカット工法による一貫ハウジング・システム

Housing-CAD (H-CAD) and its Linkage to a Total Housing System for the Pre-cut Method of Construction

横井 時人

要約 住宅業界は、活況な中で有数企業への集約化、工場での生産加工化が進み、これに伴ないコンピュータへの関心も高まりつつある。すでに、開発されて5年になる住宅CADシステム(Housing CAD, 以降H-CADと呼ぶ)は、住宅業務の流れに沿って情報が作られ、出力できる一貫システムを特長としている。H-CADのデータベースが、各業務の中で有効に適用できるような構造を持っているからである。

そのデータベースはコンピュータ内部に、屋根・外壁などを「部位」展開し、窓・流し台などを「配置」することにより、3次元の家モデルを忠実に作り上げるといった考えを持っている。

本稿では、H-CADおよび検索システムを始めとする周辺サブシステムが、各業務の中でいかに適合されるかについて述べ、さらにそれらのシステム構成、データ構造についても説明を加える。

とくにH-CADで作られる躯体を中心とする3次元家モデルが、木造軸組工法でのCAMシステム(プレカット・システム)へいかに連動されるかについて詳しく述べる。

Abstract In the active housing industry, more and more house builders are being centralized into a limited number of leading businesses, and there has been rapid progress in prefabrication at factories. All this has given rise to an increased interest in the computer.

The Housing-CAD system (hereafter referred to as H-CAD), now five years old, provides advantages as a total system, which allow data to be produced according to the flow of jobs and also the output to be listed at the same time. This is because the H-CAD database is so designed and constructed as to be effectively used for each job.

The idea employed in the database is to let a computer make up three-dimensional housing models in a very faithful fashion by positioning roofs, outer walls and the like (elements to form a surface shape) and placing windows, a sink, etc. (elements of layout) within the computer.

This paper describes how the H-CAD and some environmental subsystems including a retrieval system can be applied to the processing of each job, and also explains how those systems are configured and how the data are structured. Special details are also given about how solid structural housing models created by the H-CAD are linked to a CAM system (for the pre-cut method of construction) supporting the house timber construction method.

1. はじめに

住宅業界は活況な中に次のような状況にあり、その中でいくつかの問題点を抱えている。

1) 企業競争の激化

- ・大企業の全国展開と地場企業の拡販戦略との激突

- ・中小工務店の下請化、および衰弱化

このような状況で、他社との競争力強化をはかるため、営業戦略、細部にわたる工法等で工夫、対策を練っている企業が多い。

2) 設計業務の多忙化

- ・居住空間、形態の複雑化、居住性能の向上が著しい。
- ・住宅部材の多種、高品質化が著しい。

受注棟数の増加も加わり、設計業務が量的に多く、また質的にも高くなっている。自社内だけでは追いつけず、設計事務所などへ外注する企業が多い。

このため、設計の標準化、図面の均質化が計れず、かつ外注費が増大しているのが現状である。

3) 技能工の不足

- ・技能工の高令化に伴なう職人不足が進んでいる。

技能工の不足からプレハブ工法を始め、工場生産、工場加工の割合が増えてきている。

木造軸組工法でも、柱・梁などの構造部材を工場ですべて加工するための機械が作られ、これを制御する加工システム（プレカット・システム）が開発されている。

4) 事務処理合理化の必要性

- ・積算、発注、入出金業務の増大。

設計図書から数量の拾出しを行い見積書を作成している。

また、これらの情報から発注、出荷、入出金管理を行うが、業務量の増大から設計図書との整合性・正確性の欠如が問題となっている。

このような状況から、川上（営業折衝支援）から川下（発注・施工以降）までの、各住宅業務で必要とする情報を、なるべく多く手前の業務から引き継ぎ、人手をかけずに処理できるシステムが必要となってきた。その実現手段としてCADを中核とする一貫システムが、以前にも増して注目されている。

2. 住宅業務の流れと必要とする出力図書

戸建住宅での営業折衝から始まり、竣工後の管理に至るまでの一般的な主業務の流れと、関連する部門間での出力図書を図1に示す。

営業折衝段階では、営業用図面、見積書を迅速に作成し、数回の折衝を重ね契約に近づける。この時の見積書は、構造材の一本一本を拾わない躯体概算の方法で算出し、作られるのが一般的である。

契約前後では、実施設計図面を作成し、確認申請、工事・発注への準備を行う。設計・積算部門は、主たる出力図書の作成部所であり、また住宅業務全体から見ても重要な位置を占めている。営業折衝から実施設計の図面作成まで、成約報酬の形態で外部設計事務所に依頼するケースも多々ある。この場合でも、見積書の作成は自社で行うのが一般的である。

発注段階では、契約客との色柄合わせが行われ、部材（とくに仕上材）の正確な品番が決められる。これを受けて資材部を中心とし販売店への発注が、また工事部門を

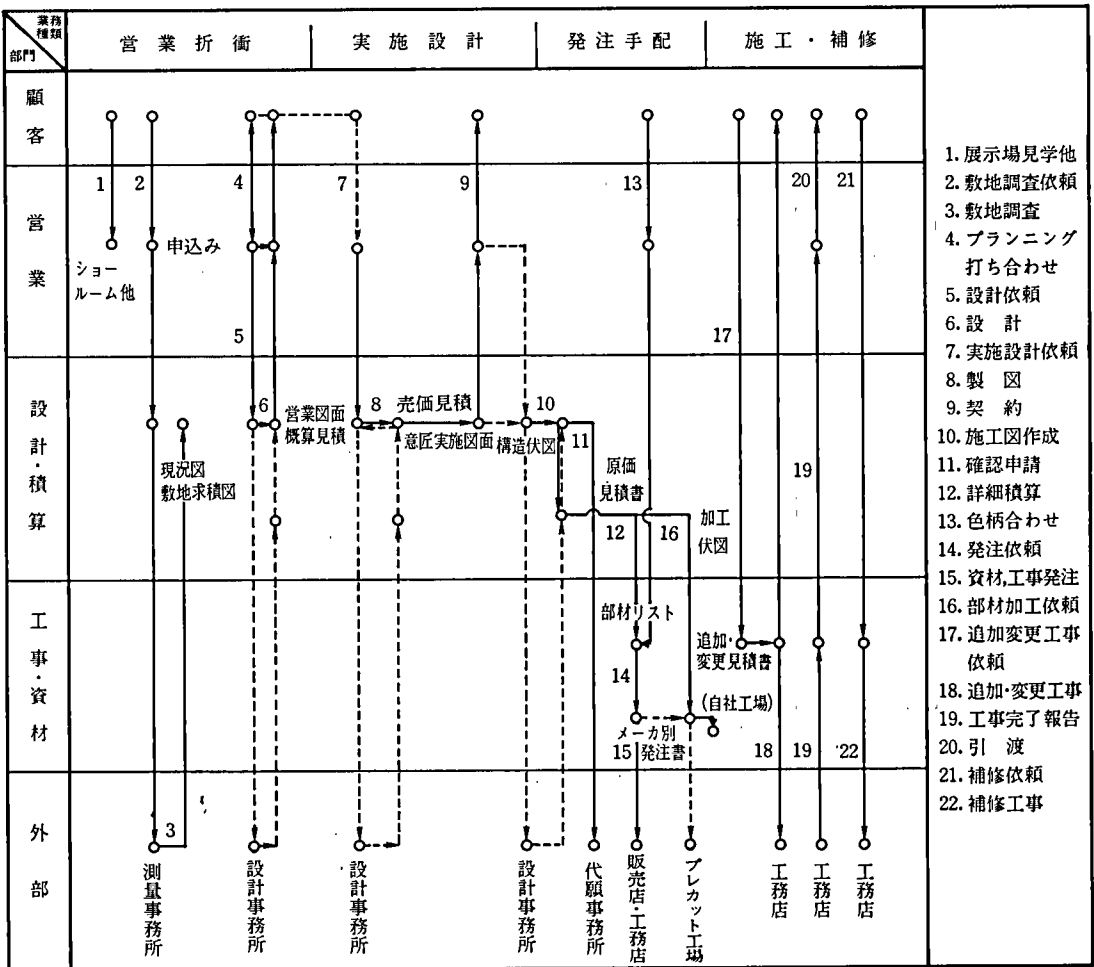


図1 戸建住宅の業務の流れ

Fig.1 Job processing of house builder

中心として工務店、各専門工事店へ外注が行われる。このとき、プレカット加工を採用している企業は、加工用伏図等の情報を工場へ渡し、プレカット加工を行っている。

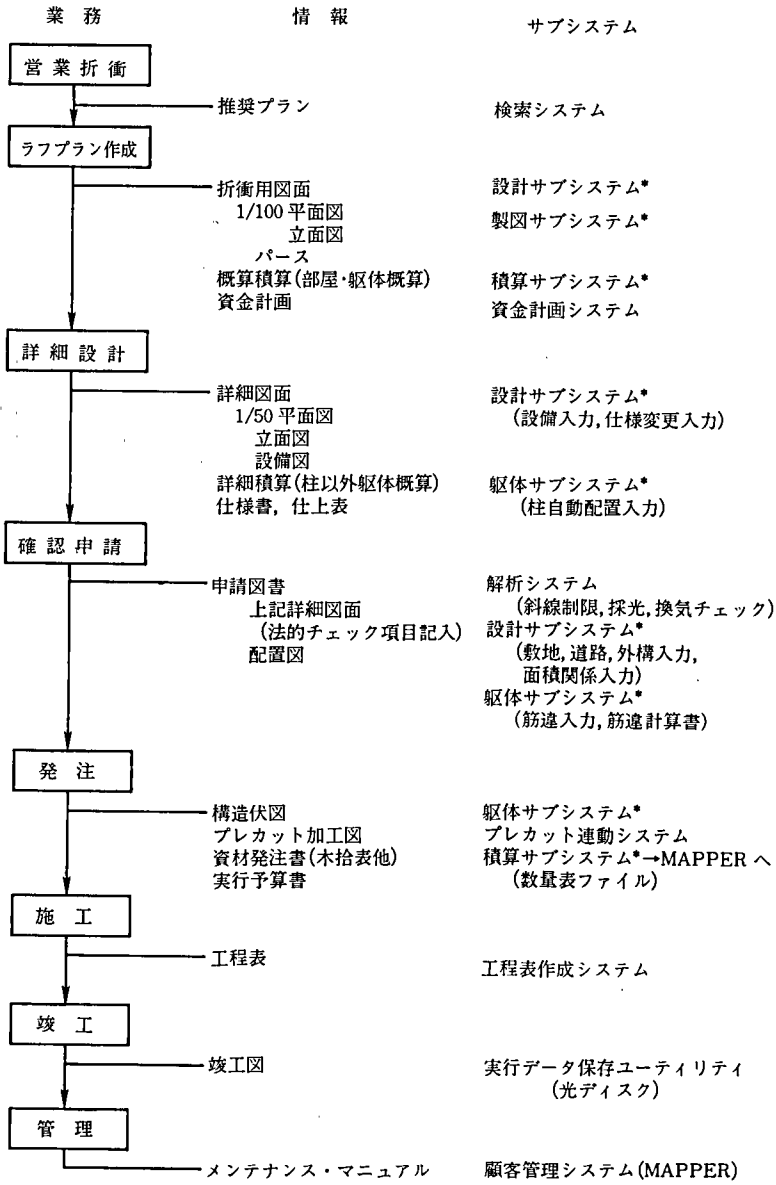
施工以降は、顧客と工事部門間での結び付きが強く、最近では補修を含め改築等、新たな営業部門を通しての市場開拓が行われつつある。

3. 住宅業務へのシステムの対応

住宅各業務の各段階で対応するシステムを図2に示す。

システム化を計るためには業務処理を明確にし、システムの運用手続きを各部所間で決めておく必要がある。また、積算指針、構造基準、取扱部材の範囲の選定、図面表現方法の統一といった社内の標準ルールを確立する必要がある。

- 1) 積算指針の作成……見積書（顧客提出用の売価見積および社内収支計算用の原価見積）に計上すべき項目、それら項目で扱う部材の数量算出方法を記述する。また、部材の単価についても材と工の区分を明確にしておく。



*印のサブシステムはH-CADとして準備されている。

図 2 業務の流れと各サブシステム出力図書
Fig. 2 Job processing and each subsystem's output document

- 2) 構造設計基準書の作成……梁、柱、土台など各種構造部材の架構ルール、すなわち種々の条件に対し、どのような構造部材をどのように位置決めしたらよいかを決める。これにより、構造図作成の標準化、施工方法の統一化を計ることができる。

また、躯体自動配置システム作成のための基本仕様書にもなる。

- 3) 取扱部材の選定……住宅で扱う部材は十万件を越すほど数多くある。顧客の要求を満足させ、かつ部材管理が可能な範囲内で、1)の積算指針に基づき取扱部材を絞り込む必要がある。

これらの部材(H-CADでは標準品と呼ぶ)は、積算、製図出力処理で用いる名称などの属性値を事前に調べておき、部材マスタ(H-CADでは標準品非形状マスタ)へ登録しておく。ここで取扱う部材は、発注で取扱う部材と一対一の対応である必要はない。一般的には代表部材であり、発注処理の段階で詳細な情報が付加されるか、または構成している部材に展開されるものである。

- 4) 図面表現基準書の作成……各種図面で表現すべき詳細表現方法および、例外・特殊表現方法を各部材の種類ごとに決めておく必要がある。

各部材が固定(伸縮は自由)の形状を持つものは、図形マスタへ事前に定義しておく。また、図形の特別処理、文字などの表現情報は、3)で述べた部材マスタへそれらの属性値を定義しておく。

このような、標準ルールを設けシステムを構築することは、省力化・合理化・外注費軽減といった数字で把握される定量効果以外に、品質の向上という数字にすぐ表れない定性効果も生み出す。

設計・積算部門で作りに出される出力図書類は、住宅業務の中で最も重要なものであり根幹である。この出力図書を作り出すCADシステムの構築によって、その企業の効率的生産合理化が達成できるかどうかが決まってくる。

4. 家モデル

中枢となるCADシステムで最も重要なことは、出力のもととなる家のモデルをどう把握するかである。多くのパソコンの住宅CADなどでは、一般的な出力図面に対し単一指向での家に関するデータを、直線や円弧といった図形データの集合として扱っている。また、それら個々のデータに付加情報として、見積書作成など後処理に必要な属性値を持たせている。このようなデータの持ち方のシステムは、一般に製図システムと呼ばれている。

一貫した情報システムを考える時は、業務の流れに沿って情報が付加され増殖され、任意な時点で必要な出力が得られるデータ構造を持たなければならない。そのためには、家を構成するものを考え種類分けし、ものをそのままデータベースに持つ方法がとられる。

住宅は一品生産であり、すべての形態(居住空間)が異なる。そして、その形態に配置される部材は多種多様である。家を構成する情報を自由に定義でき、家をコンピュータ内部に忠実に表現できることが重要である。

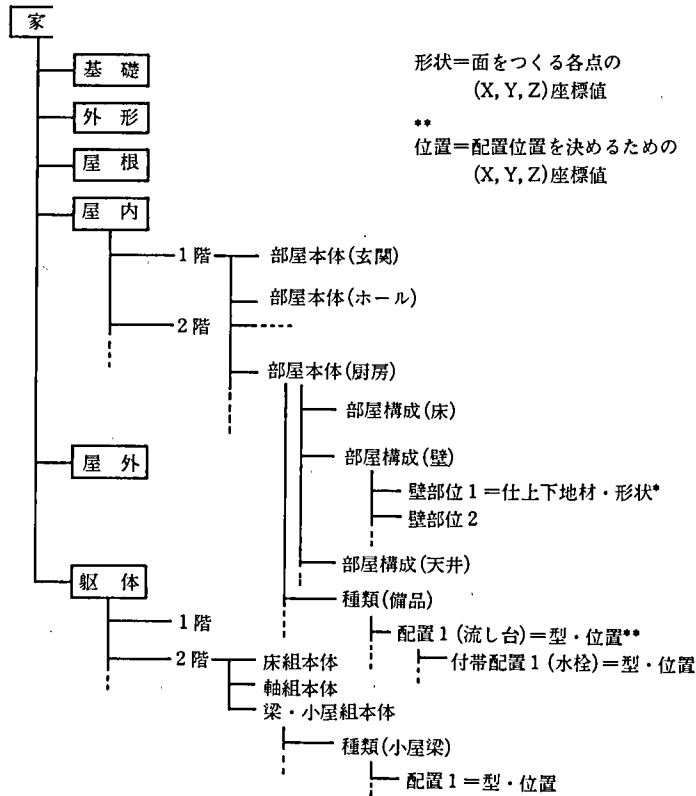
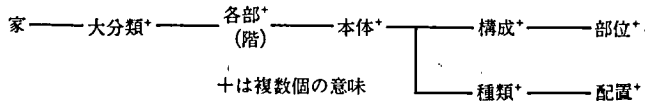


図 3 家モデルのデータベース構造
Fig. 3 Database structure of a housing model

H-CAD における家モデルは、一棟分の家を構成する各種の標準品を分類し論理的に構造化したものである。この構造の中では、自由に立体平面を作る“部位もの”と既製形状を持つ“配置もの”に分けられる。

家モデルのデータベースは、階層構造を取っており次のような体系を持つ。



この構造に従い、操作によって選ばれた標準品データは、操作者が指示する形状、あるいは配置される位置が付加されて、関係データベースによりデータ間で相互の関係が持たれ家モデルが作られていく。

5. システムの概要

図 2 で示した業務の流れにそって、使用される各システムの概要を述べる。

5.1 検索システム

1章で述べたように設計業務は多岐にわたり、設計技術者でなくても営業折衝段階でのプランニングをできないか、という要求が出てきている。一般の顧客の場合、敷地の大きさ・地形・家族構成・希望建物面積などで、おおむねの間取りは決まると言

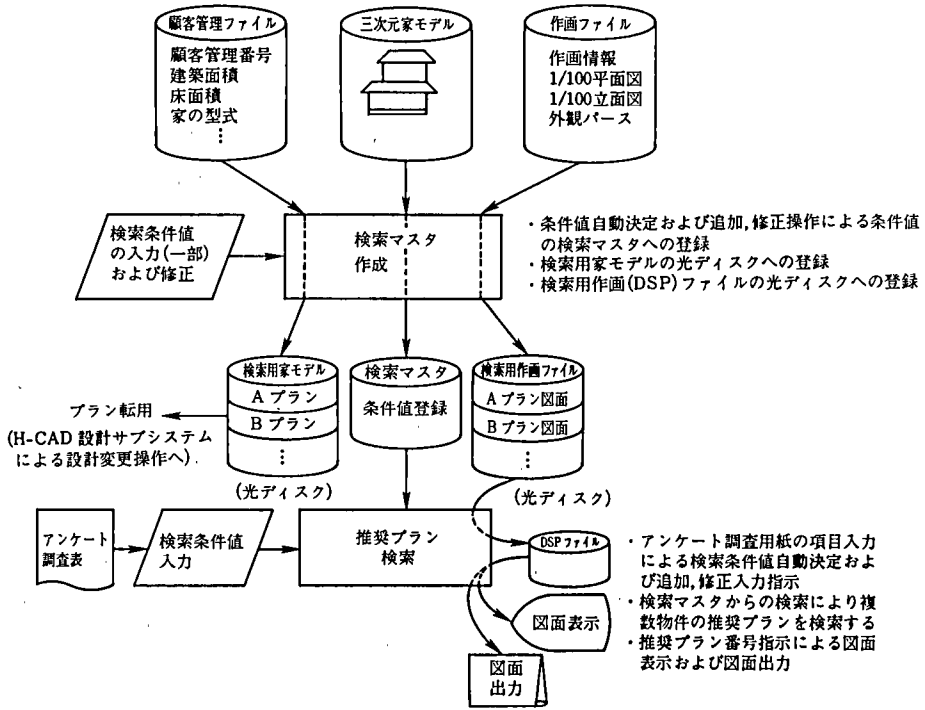


図4 検索システムの構成

Fig. 4 Overview of the housing retrieval system

われている。しかし、このような条件で顧客の希望イメージに合う間取りプランを検索し、推奨するためには約1万5千から2万プランを検索マスタへ登録しておく必要があるとされている。このような要件をもとに、検索システムの概要を述べる。

- 1) システム構成……検索システムは、検索マスタへ条件値を登録する検索マスタ作成と、顧客の希望条件から推奨プランを検索する推奨プラン検索の二つから成り立つ。これらのシステム構成は図4の通りである。
- 2) 検索マスタ作成……検索マスタ・ファイルを作成するためには、プランナンバごとの各検索条件項目に対する条件値を自動決定、または入力し登録する必要がある。

検索条件項目としては、表1のようなものが考えられる。

表1 検索条件項目
Table 1 Samples of retrieve key

- ・延床面積サイズ
- ・建物の間口×奥行サイズ
- ・居室総数
- ・台所・食堂・居間の合成タイプ
- ・玄関の取付方向タイプ
- ・和室総数および配置タイプ
- ・厨房の位置タイプ
- ・子供部屋数
- ・吹抜有無
- ・階段タイプ、等

これらの中には、H-CAD が持っている顧客管理ファイル、および設計サブシステムで作られた三次元家モデルから自動的に条件値を決定させ、登録させることができる項目もある。

検索マスタ・ファイルを作成すること以外に、登録後のプランを画面に表示したり図面出力が可能ないように、作画 (Drawing Subroutine Package: DSP) ファイルを、取り出しやすい分類のもとに保存しておく必要がある。この作画ファイルは、大容量のものとなるため光ディスクへの格納が適当である。

また、検索後の推奨プランをプラン転用し、変更して顧客の物件にすることを考える時には、三次元家モデルも保存しておく必要がある。しかし、登録すべき検索対象物件が多い時は、その保存ファイルが作画ファイルに比べ、さらに大きい容量を必要とするためコストがかかる。作画ファイルから図面出力し、それをもとに新規入力を行う方法をとる時は、家モデルの保存ファイルは不要である。

- 3) 推奨プラン検索……建築見込客は、当初は具体的プラン計画を持っていない(または、住宅メーカーへ教えない)。顧客がすぐ記入できる、または営業マンがすぐ調べられる情報から、住宅イメージを引き起こすことができたならば強力な営業支援となる。展示場、ショールームで行われるアンケートの内容は次のようなものである。

-
- お手持の敷地面積は。
 - 道路はどちら側にありますか。
 - 建築予算は、坪単価は。
 - 家族構成は。(子供○人、老人○人、…)
 - お仕事は。
 - 失礼ですが、奥様の年齢は。
- ⋮
-

これらのアンケートの内容から、2) で述べた各条件項目の値を推察して自動決定できればよい。これら条件値のもとで、推奨プランが検出できない時は、順次、条件値の範囲を段階的に拡げていき検出できれば便利である。このようなプログラムの開発には、知識システムの応用が考えられる。

5.2 H-CAD システム

H-CAD システムは、大きく五つのサブシステムから成り立っている(図5)。これらのサブシステムは、いかなる時点でも自由に選択し実行することができる。すなわち、住宅業務の流れの中で自由に家モデルに対し、登録・変更・追加を行い、その時点で必要とする出力情報をタイミングよく得ることができる。

- 1) システム構成……サブシステムは、家を構成するものを事前に標準品マスタに定義しておくための標準品サブシステム、操作により家モデルを作り上げる設計、躯体サブシステム、図面を出力する製図サブシステム、積算図書を出力する積算サブシステムがある。これらのシステム関連図を図5に示す。
- 2) 標準品サブシステム……家を構成する各種部材(外形・屋根・部屋・建具・造作・備品・設備・基礎・構造部材など)は、あらかじめ標準品データとして属性を定義しておく。属性は非形状属性と、形状属性に大別される。非形状属性には、次のようなものがある。

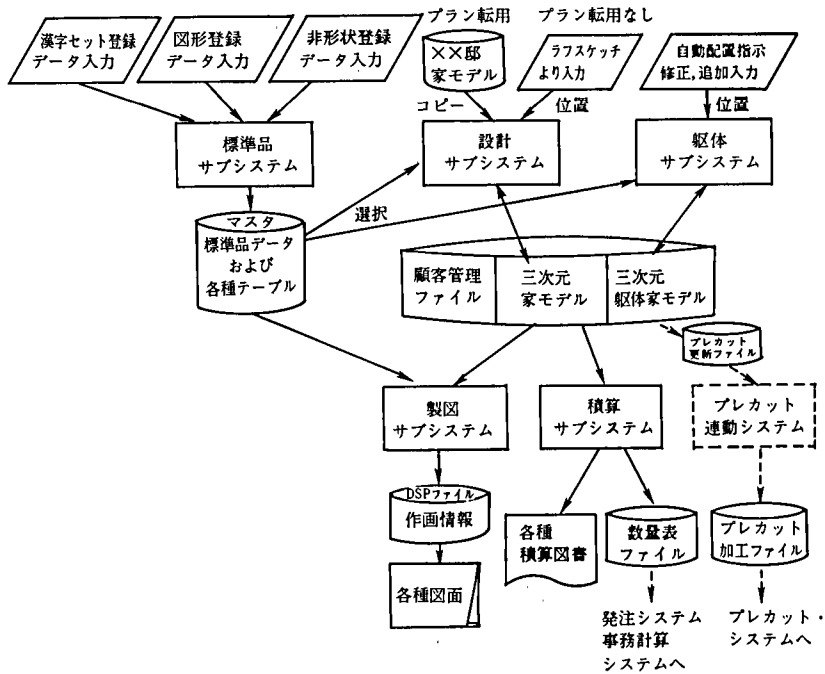


図 5 H-CAD のシステム構成

Fig. 5 Overview of Housing CAD system

- ・名称類 (名称・メーカー名・型番・色)
- ・入力情報 (位置, 形状を決めるための各種属性)
- ・図面に表現する注記 (文字) 情報, および図形番号 (配置もののみ)
- ・積算で必要とする積算処理情報
- ・ものを決めるための型決定属性
- ・標準品の種類ごとに異なる個別属性
- ・各サブシステムの自動創成, 解析などで必要とする属性
- ・子部材, 部材展開に必要な付帯標準品属性

形状属性とは、図面・画面に表現するための図形のことであり、三次元形状と投影面（二次元）形状の2種類がある。これらの図形は座標値に変数名を用い、伸縮自在なパラメトリック形状を定義できる。

- 3) 設計サブシステム-----標準品データをメニューから選択し、ラフスケッチをかけたタブレット上で、形状または位置を指示していく。これらの操作により、コンピュータ内部には、4章で述べた三次元家モデルが作り上げられていく。
このサブシステムで作られる家モデルは、構造用骨組材（躯体）を除いたものである。
- 4) 躯体サブシステム-----構造用骨組材の躯体家モデルを作り上げていくためのシステムである。構造用骨組材は、屋根の垂木、床組の根太のようにある決められた領域内に等間隔で配置されるもの、間柱のように柱材の位置を除き壁線分に等間隔で配置されるものがある。このようなものに対し、領域、線分を指示することにより、指定されたピッチでグループ配置を行う機能がある。

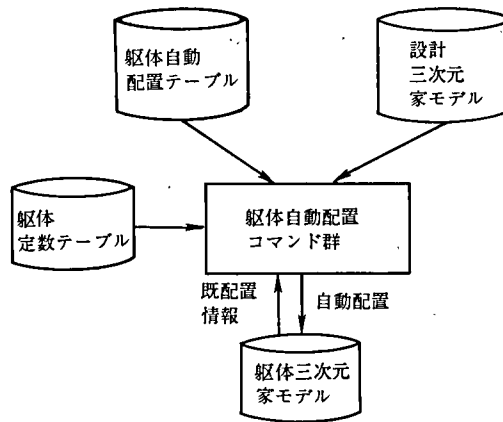


図 6 躯体自動配置システムの構成

Fig. 6 Overview of an automated structural system

しかし、これら入力のための伏図のラフスケッチ作成は、構造の技術的知識を持った人でないと困難である。また、たとえ技術者が知識を持っていても構造設計は人によって異なるのが実情である。

そこでH-CADでは、各住宅メーカーの構造設計基準に従い、各種構造部材を自動配置できるシステムをコマンド単位に付加できるようになっている(図6)。

木造軸組工法での、手入力指示、自動配置コマンドを組み合わせた一般的な躯体家モデルの操作手順を図7に示す。

- 5) 製図サブシステム……平面図・給排水設備図・電気設備図・各種構造伏図など、数多くの伏図を作成することができる。それらの図は、各種単位図(レイヤ)の組み合わせ(重ね合わせ)により作られる。たとえば平面図は単位図として、平面意匠図・備品設備位置図・家具配置位置図・電気設備位置図・柱伏図などを重ね合わせて作られる(図8)。これらの重ね合わせは、テーブルで自由に制御することができる。

伏図以外に、立面図・外観パース・室内パースなども出図することができる。これらの図は、関係データベースで作り上げられた家モデルを効率よく検索し、自動製図処理される。出力図面は、人間の加筆操作が少なければ少ないほど完成度は高い。寸法線、注記など、なるべく自動出図できる部分は内部処理を行っている。

- 6) 積算サブシステム……現在の営業段階での見積方法は、まず標準仕様での m^2 単価に対し顧客の設計面積を掛け標準仕様価格を算出する。それに、標準仕様外の標準品の価格を加減算して見積全額を出すという方法が多い。しかし、この方法では顧客にとって明細がわかりにくく工事発注では、ものが正確に把握できず再度拾い出しをしなければならないという問題がある。

H-CADの数量算出方法には次の三つのタイプがある。

- ① 家モデル内の部位形状、“配置もの”から数量を算出する(子部材の算出も含む)。

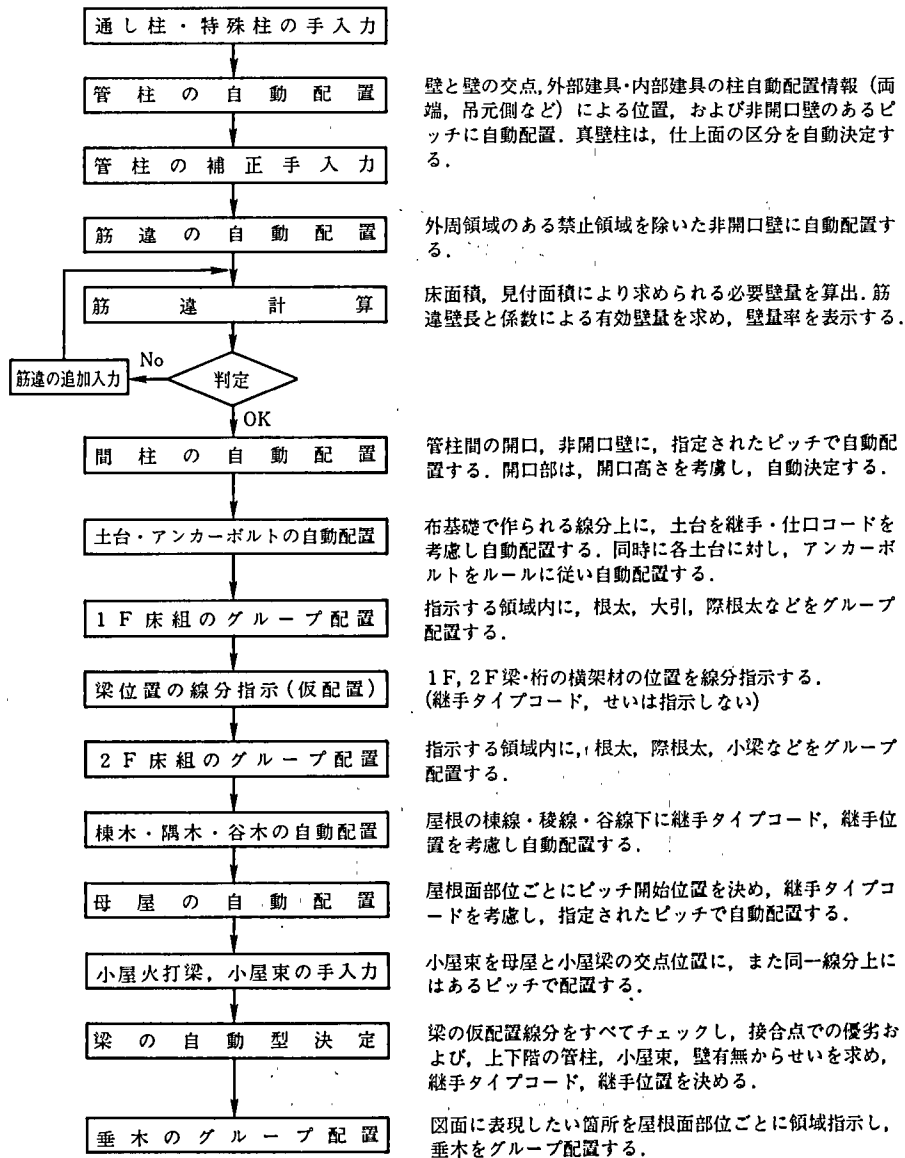


図 7 木造軸組工法での躯体操作手順

Fig. 7 Operation sequence of structural design system for a timber house construction

- ② 家モデル内のものから、ある決められた数量計算方法によりプログラム処理を行い、決められた番号(Cal #と呼ぶ)へ求めた数量を格納する。
 (例) 一階基礎内面積、軒先長さ等
 見積書で計上したい項目(例、仮設工事、板金工事等)によっては、これら Cal #を指定し、求められた数量を取り出し出力することができる。
- ③ 家モデル内に持てない見積書の項目に対し、操作者が直接、数量・単価を与えるもの。
 (例) 外構工事、解体工事、等……

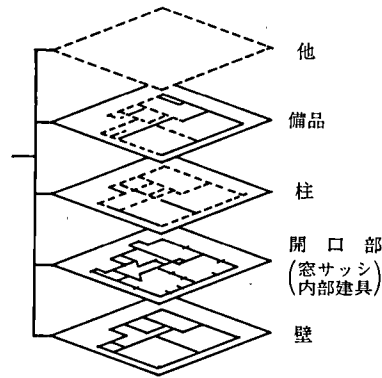


図 8 平面図のレイヤ構成

Fig. 8 Layer component of floor plan

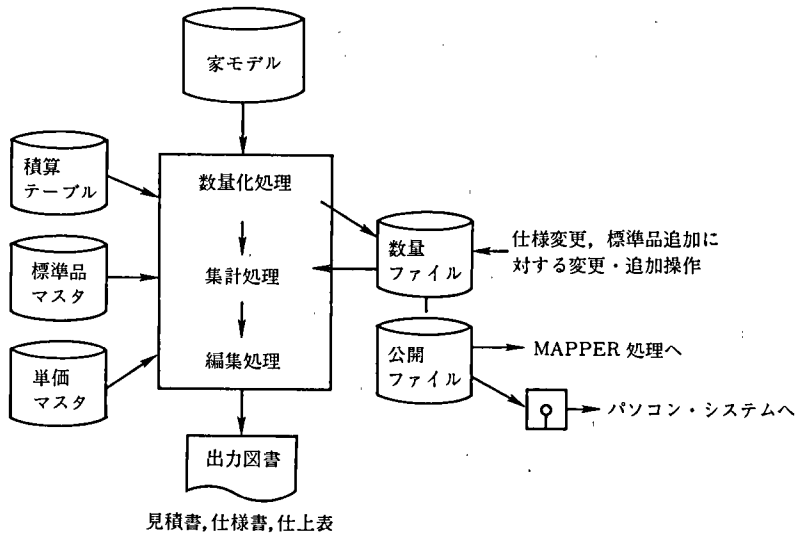


図 9 積算サブシステムの関連図

Fig. 9 Overview of the estimation subsystem

一棟分で求められる数量は、①が大半であるが、これらの数量から本体単位で数量表ファイルとして中間ファイルが作り出される。このファイルは図書変更操作により、自由に名称、数量、単価などを修正・追加ができる。とくに、非形状標準品マスタに存在しないものを追加する時や、色などの具体的仕様が決まった時に有効である。

この数量ファイルから、見積書・仕様書・仕上表が編集される。また、このファイルは各住宅メーカーで作られた MAPPER による受発注システム、パソコンによる差額見積書作成処理などで使用されている (図 9)。

5.3 プレカット連動システム

木造軸組工法では主要構造部材として、柱、土台、横架材の梁・母屋・棟木などがある。これら部材端部の継手、四側面での仕口加工は高度な技能を要し、また手間もかかる。そこで、切削加工による自動加工機械が作られ工場ではプレカットが行われるようになった。最近ではパソコンで加工情報を作成するシステムもある。この情報は、

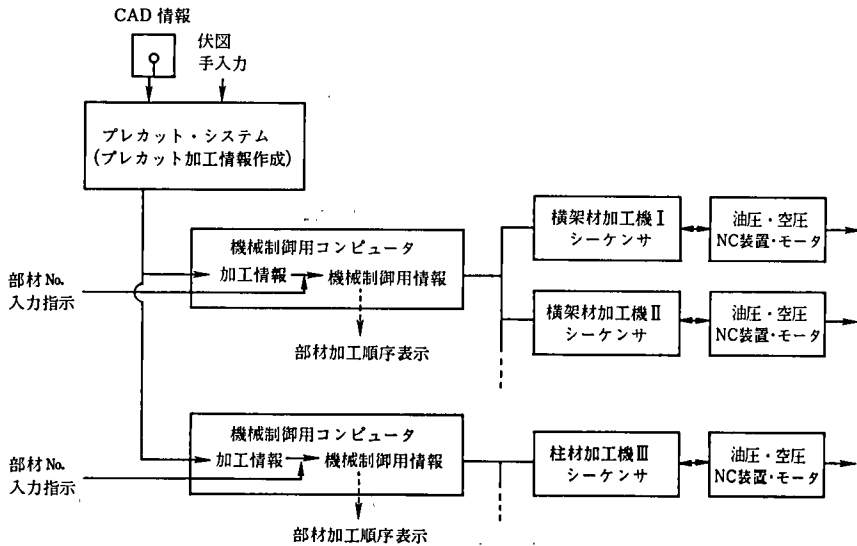


図 10 プレカット加工ライン構成図

Fig.10 Pre-cut processing line

工場側の機械制御用コンピュータへディスク経由で、または直接回線経由で渡される。工場側機械では管面に表示された操作メニューにより、加工部材ナンバを予約し自動加工起動スイッチを押すことにより順次加工していく。

加工情報は機械制御用情報に変換され、機械各部の NC 制御装置、油圧、空圧装置、モータ類のシーケンス制御を行い、構造部材の加工がなされる (図 10)。

ここでは、H-CAD で作られた CAD データが、加工情報を作り出すプレカット・システムへどのようにして引き継がれるかについて述べる。

- 1) 木材加工内容……機械が木材を加工する内容は、①長さを決め両端を切断、②両端の継手加工、③四側面のほぞ穴・仕口加工である。

木材加工の流れは加工機械の仕様により異なるが、横架材について代表的加工メーカー 2 社の例を図 11, 12 に示す。

- 2) 横架材各加工機械へ渡すための CAD で必要な属性

- ① 長さ決めカット

- ・両端の芯-芯間長さに、両端の継手タイプにより決まる長さ補正値を加えた実長。

- ② 側面加工

- ・自材の幅、せい
- ・上下(柱、間柱、小屋束など)、水平直交部材の交点位置および相手部材の継手タイプ、幅、せい

- ③ 継手加工

- ・自材の幅、せい、継手タイプ
- ・端部で同一水平方向に接続する場合の相手部材の継手タイプ、幅、せい

- 3) プレカット連動……H-CAD の設計、躯体サブシステムで作られる家モデルか

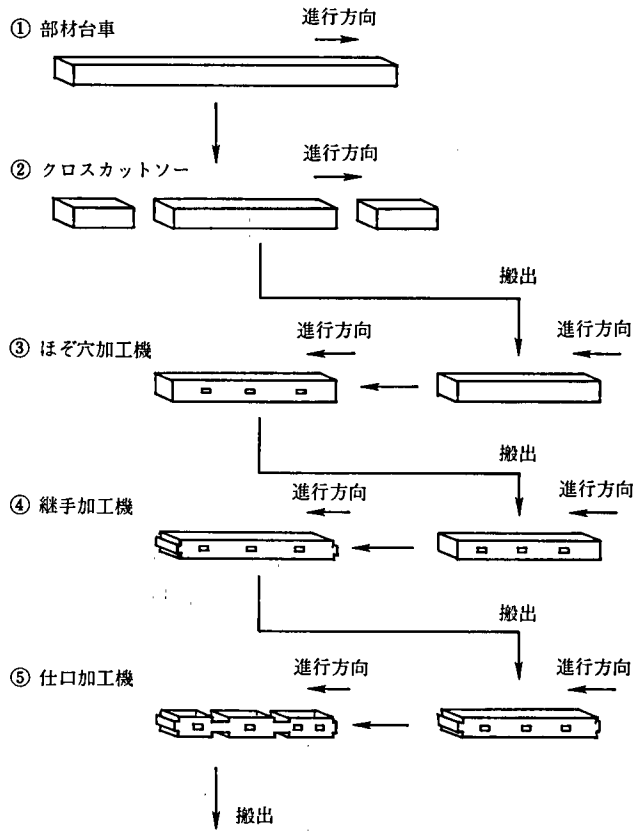


図 11 H 社の横架材加工

Fig. 11 Pre-cut processing of a horizontal timber by H Ltd.

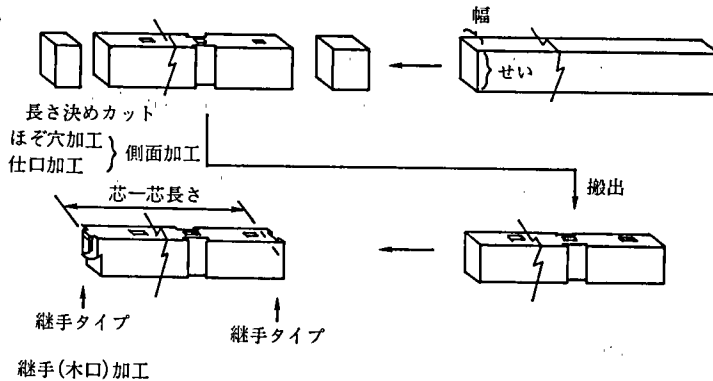


図 12 M 社の横架材加工

Fig. 12 Pre-cut processing of a horizontal timber by M Ltd.

ら、2)に述べた加工情報を作り出すために必要な情報を取り出し、ファイルとして公開している。各社プレカット・システムに合うように、この更新ファイルを変換してプレカット・マシンへ連動することができる(図13)。

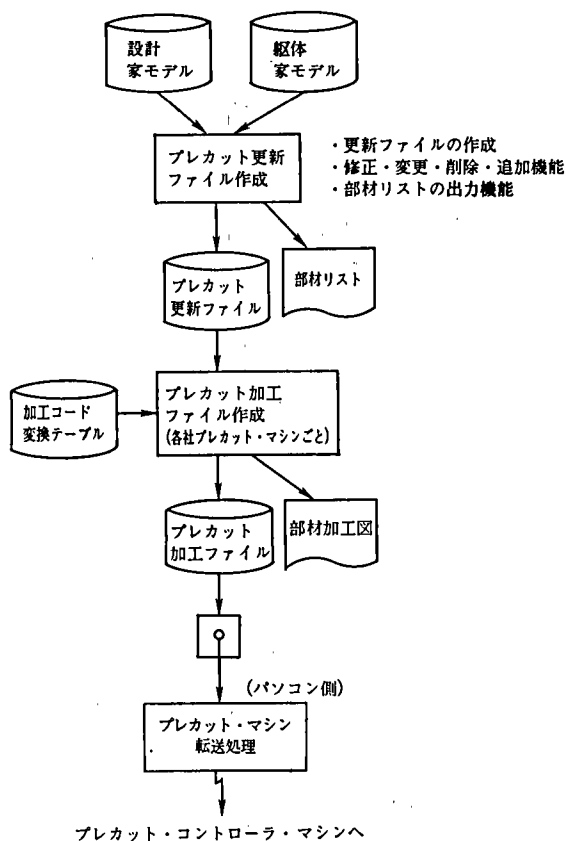


図 13 H-CAD プレカット連動システムの関連図

Fig. 13 Overview of the Housing CAD connected to the pre-cut system

4) プレカット更新ファイルの内容

- 見出し情報=管理番号・顧客名・工事名・階数
- 対象部材情報*
 - 見出し=部材コード・階コード・全部材本数 (n)
 - 部材情報=始点位置・終点位置・始点継手タイプコード・終点継手タイプコード・幅・せい・芯-芯間長さ・実長・材質・等級
- 付加情報
 - 部屋情報=部屋タイプコード・領域・床高・天井高
 - 開口情報=開口タイプコード・開口高・始点位置・終点位置
 - その他*
 - 屋根
 - 外形

部屋、開口などの情報は、主に柱材の加工（四側面への貫穴加工など）のために用いられる。

このファイルから情報の内容を印書することができる。

5) プレカット・システム-----プレカット・システムは、H-CADのプレカット更新ファイルを読み、プレカット加工情報を作成するシステムである。図 14 に概要フローの例を記す。

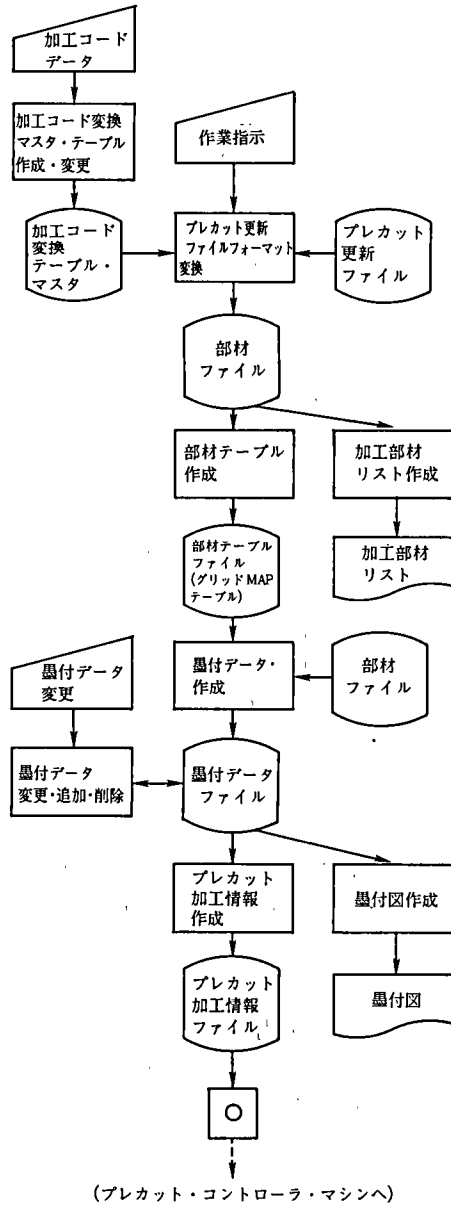
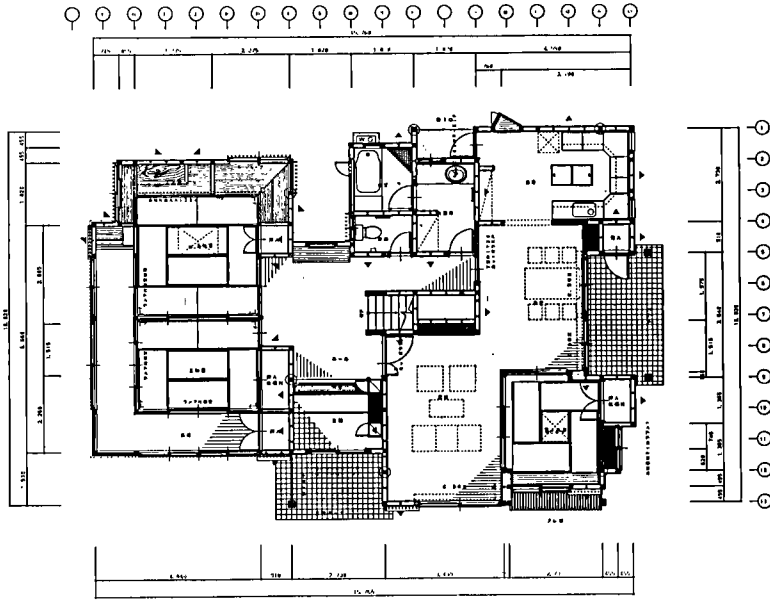


図 14 プレカット・システム概要フロー
Fig. 14 General flowchart of a pre-cut system

6. おわりに

H-CAD が住宅業務の中で核となり、一貫システムが構築できることを述べてきた。家モデルを家を構成するものでとらえ、設計操作時に三次元で家を作り上げたならば、必要とする図面・見積書・数量表などが自動出力されるシステムである。今後は、それらの出力図書の自動化率を高めていく必要がある。また、コンピュータ・グラフィックス、エンジニアリング・ワークステーション、図面読取装置などハードウェアの技術進歩をうまく応用し、住宅分野での先駆的 CAD を今後ともめざしていきたい。

〔付 録〕 H-CAD による木造軸組工法での出力例



(a) 1階平面図



南立面図



パース方向4外観透視図

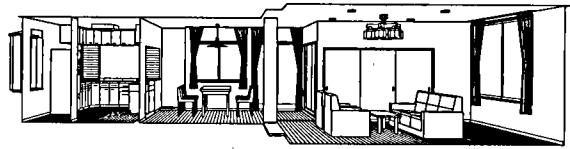
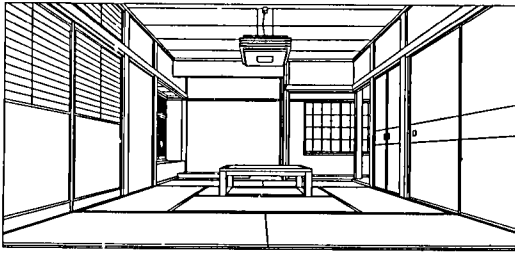


西立面図

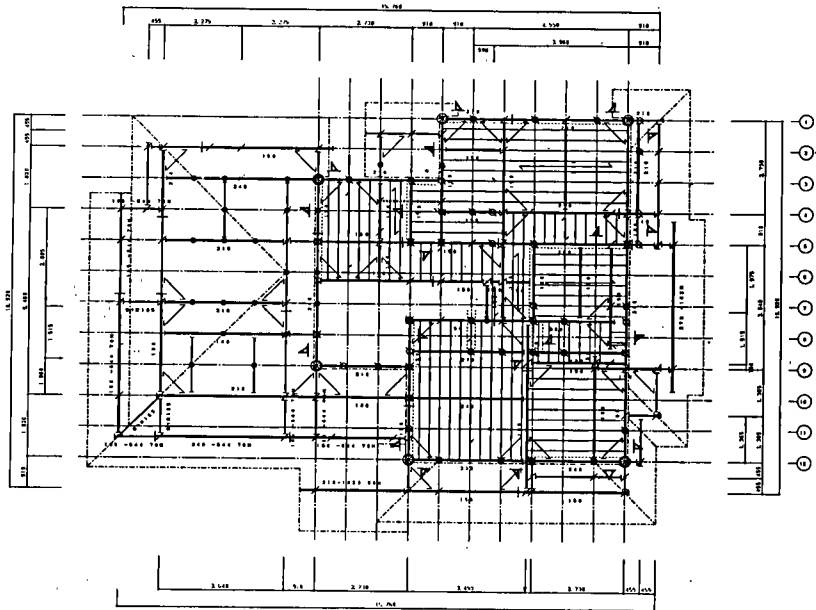


パース方向1外観透視図

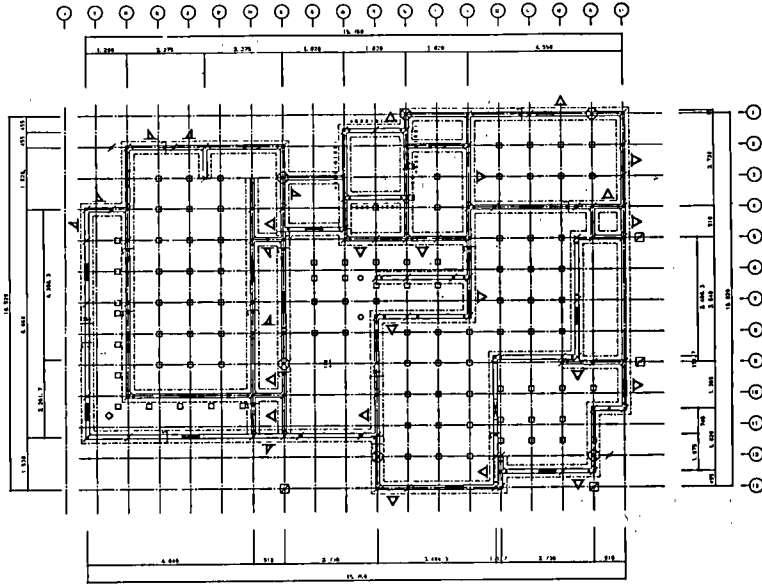
(b)



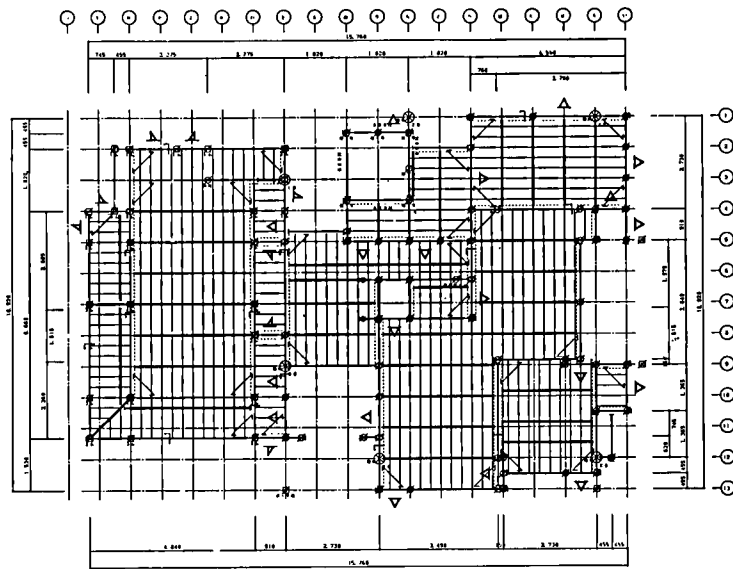
(c) 室内透视图



(d) 2階床伏图・1階小屋伏图



(e) 基礎伏図



(f) 床伏図

-
- 参考文献 [1] “建築生産方式のシステム化”，建築生産情報の統合化に関する調査研究報告書，建設省建築研究所，昭和61年3月。
- [2] 熊沢孝司，“プレカット機械の現状と展望”，住宅と木材，日本住宅・木材技術センター発行，昭和61年12月。
- [3] “住宅建築の実施例”，CAD/CAM 事例集シリーズ5，技報堂出版。
- [4] 横井時人，“木造軸組工法とCAD”，第三回ハウジング研究報告集，(社)日本住宅協会，昭和62年3月。
- [5] 渡辺寛，“枠組壁工法住宅における構造の自動設計システム”，日本ユニバック技報，1988年2月，第16号。
- [6] 篠田博水，“第3次元家モデルに基づく住宅一貫システム”，日本ユニバック技報，1985年2月，第8号。

執筆者紹介 横井時人 (Tokihito Yokoi)

昭和43年千葉大学工学部建築学科卒業。45年日本ユニシス(株)入社，技術計算，建築アプリケーション・ソフトウェアの開発を経て，ハウジングCADシステムの開発，SEサービスに従事。現在，関西支社システム一部CAD/CAM2課に所属。日本木造住宅産業協会技術開発委員会幹事，日本建築学会員。



鋳物凝固解析プログラム CAST のアルミニウム合金鋳物への適用

The Application to Aluminum Alloy Castings by a Solidification Analysis Program CAST

松 林 幸 枝

要 約 本稿では、3次元鋳物凝固解析プログラム CAST (CASTing analysis program) の紹介を行う。CAST は有限要素法に基づき、鋳物充填時より全体の凝固に至るまでを凝固収縮流を考慮して解析を行う。

今回は、これまで機会のなかったアルミニウム合金鋳物の凝固過程を CAST により解析し、実験結果との比較検討を行った。

Abstract Three dimensional solidification analysis program CAST is introduced in this paper. CAST is based on Finite Element Method and takes account of fluid flow induced by solidification shrinkage.

Several solidification process of aluminum alloy have been analysed and compared with experimental results.

1. はじめに

鋳物製品に対するコスト低減と品質の向上、とくに鋳造欠陥防止への要求が強く、解析技術の利用によるひけ巣予知法が近年多く試みられている。有限要素法による3次元鋳物凝固解析プログラム CAST を開発し、これまで機会のなかったアルミニウム合金鋳物への有効性を検討した。

今回のプログラム評価作業にあたり、鋳型製作、温度測定など実験の上で、光生アルミニウム工業株式会社の多大な協力を得た。アルミニウム合金での解析例は、比較的例が少なく実験との詳細な比較データに欠けていることから形状の簡単な数モデルを選び、評価目的を次の2点とした。

- 1) 解析冷却曲線と測定冷却曲線が合致するようなパラメタを決定し、今後のアルミニウム合金鋳造シミュレーションの基礎をつくる。
- 2) 解析から得られるひけ巣予知情報と鋳造品切断面の比較による予知情報信頼性の確認をする。

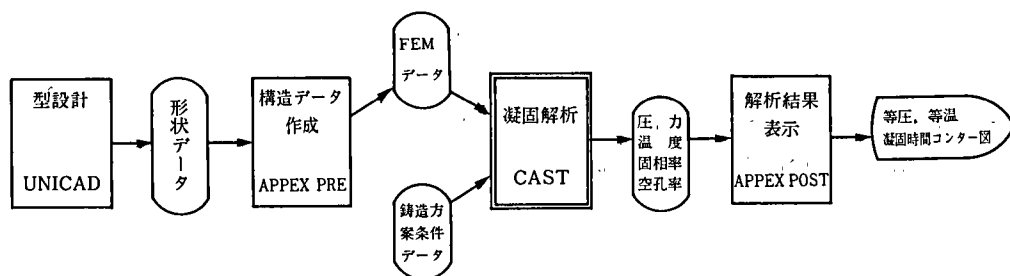
2. 凝固解析プログラム CAST の概要

CAST は凝固収縮流を考慮した有限要素法による3次元凝固解析プログラムであり、大中らによる論文¹⁾(内節点法を採用)を有限要素法により実現したものである。周辺ソフトウェアを含めたシステム構成例を図1に示す。

2.1 プリプロセッサ

凝固解析は、必然的に複雑形状の3次元中実モデルを対象とするため、入力データ作成の容易さが解析の鍵となる。汎用構造解析プログラム NASTRAN が多分野に広く浸透しており、NASTRAN への入力形式は FEM 入力の標準といえる。

CAST の入力データは NASTRAN 入力データ形式を採用しており、当社のプリプ



UNICAD 汎用3次元CAD/CAMシステム
 APPEX1100 汎用FEMプリ・ポストプロセッサ
 (PRE, POST)

図1 システム構成

Fig.1 System configuration

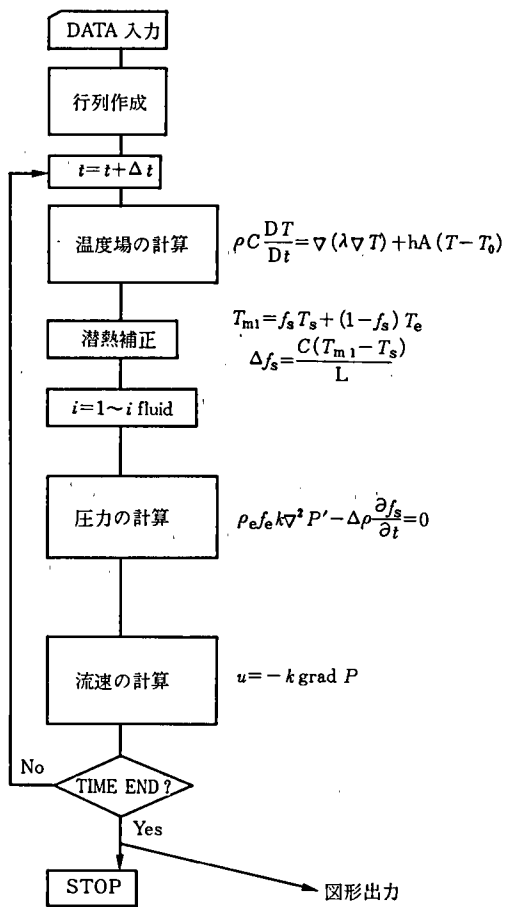


図2 計算の概略

Fig.2 System flow of analysis

ロセッサ APPEX が連動する。また、すでに設置されているであろう NASTRAN 用データ作成プログラムはそのまま使用できる。

2.2 解析プログラム CAST

採用した計算手法は先の論文^[1]に詳しく掲載されているので、以下にその要約をあげる。図 2 は計算の概略を示す。

- 1) 有限要素法により、エネルギー保存則、質量保存則、運動量保存則を連成して計算する。
- 2) 固液共存域、および液相域での液相の流動はダルシーの法則に従う。
- 3) 凝固潜熱の放出は温度回復法により考慮する。
- 4) 各種物性値（熱伝導率、熱伝達係数、比熱、凝固潜熱など）の温度依存入力および温度、圧力拘束の時間依存入力が可能である。
- 5) 積分時間刻み幅を内部でコントロールする。

2.3 ポストプロセッサ

3次元中実体では解析結果の図形表示がとくに不可欠である。入力形式同様、出力形式も NASTRAN 形式を採用しているため、APPEX/POST や他のポストプロセッサにより、多様な図形表示が可能である。

- 1) 指示節点の冷却曲線（温度-時間）
 - 2) 指定時刻での任意断面の等温線
 - 3) 指定した限界固相率での等凝固時間線
 - 4) 指定した限界固相率での等温度勾配 G 線
 - 5) 指定した限界固相率での等 G/\sqrt{R} 線 (R は冷却速度)
- などがカラー表示され、ひけ巣位置予測をたすける。

上記各線図については 4 章で説明する。

3. アルミニウム合金鋳物への適用

3.1 実験および解析モデル

使用材料はアルミニウム合金が AC-2A、鋳型が S45C、塗型がダイコート 34 である。回転軸対称体とプレーキシリングの形状を口絵 1 と口絵 2 に示す。前者は鋳型を含めた軸対称 2次元解析とし、断熱系塗型吹付無しをモデル 1、吹付有りをモデル 2 とする。プレーキシリングは金型温度を固定境界とした 3次元ソリッド解析でありモデル 3 とする。回転軸対称体について、軸対称 2次元解析と 3次元ソリッド解析を同一物性値を使って行い、同一の解が得られることを確認した。

各々のモデルに採用した物性値を表 1 に示す。これら数値は、後述するように実験と解析冷却曲線を近づけるために、多くの試行を繰り返したがその最終結果である。

3.2 温度計測

口絵 1 と口絵 2 に示す位置を計測点とし、熱電対により同時 3点測定をした。鋳込み温度は $700^{\circ}\pm 10^{\circ}\text{C}$ であり、型温はバーナにより約 300°C に予熱した。注湯と同時に鋳物温度が急上昇するためレコーダの針が追従できず、最高温度に大きなバラツキが見られ、数回の試行からほぼ妥当と思われる温度曲線を採用した。また、鋳物径が小さいモデルのため、熱電対リード線が注湯時に影響があったと考えなければならない。

表 1 解析モデルに採用した物性値

Table 1 Material properties used in analysis

(材料定数 単位 :cal, mm, sec, g)

アルミニウム合金	固相	共存相	液相
熱伝導率 λ [cal/mm sec deg]	0.05	0.026	0.020
密度 ρ [g/mm ³]	2.79×10^{-3}		2.5×10^{-3}
比熱 C [cal/g]	0.35	0.25	0.2
凝固潜熱 L [cal/g]		30~20	
温度 T_s, T_L	526°C	~	604°C
初期温度 [°C]		700°C	
熱伝達係数 h	1.3×10^{-4}	~	0.9×10^{-4}
(鑄型間) [cal/mm ² sec deg]	4.0×10^{-5}	~	モデル 1 (塗型無) 5.0×10^{-5} モデル 2 (塗型有)
鑄型			
熱伝導率	0.0106		
密度	0.0078		
比熱	0.11		
初期温度	260~280°C		

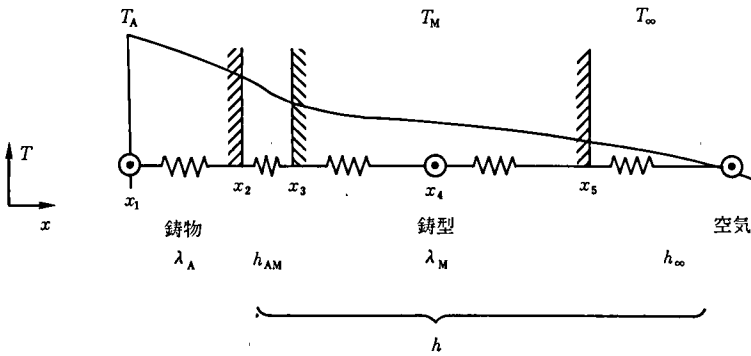


図 3 断面の熱抵抗

Fig. 3 Heat resistance of section

3.3 解析条件設定の問題点

3.3.1 パラメタ設定

1) 熱伝達係数 h ……“金型鑄造において凝固に及ぼす影響の最も大きいものの一つに鑄物—金型間の熱抵抗がある”^[2]。

解析では、鑄物—鑄型間の境界熱伝達係数、鑄型—空気間と鑄物—空気間の環境熱伝達係数の設定が必要であるが、とくに境界熱伝達係数は大きな影響を及ぼす^{[3][4][5]}。

境界熱伝達係数は鑄物—鑄型間接触状態（エアギャップ発生の有無による放射冷却、伝熱あるいは断熱塗型の効果等）により著しく変わりその数値設定は困難となる。

図3のように、鋳物温度 T_A 、鋳型温度 T_M 、外部温度 T_∞ が測定できれば境界熱伝達係数 h_{AM} と環境熱伝達係数 h_∞ の関係は下式となる。

$$\frac{1}{h_{AM}} = \frac{T_A - T_M}{T_M - T_\infty} \left(\frac{1}{h_\infty} + \frac{X_5 - X_4}{\lambda_M} \right) - \left(\frac{X_2 - X_1}{\lambda_A} + \frac{X_4 - X_5}{\lambda_M} \right)$$

測定精度の問題もあり、 h_{AB} のおよそのオーダーを決めるのに使った。

3次元複雑形状では、鋳型のモデル化が困難であること、鋳型モデルを含めることによる解析実行時間が急増することから、現状では鋳型を含めない解析をすることが多い。モデル3のブレーキシリンダはその例であり、鋳型部分を熱伝達係数だけで表現する。鋳型部分を代表する熱伝達係数 h のオーダーを次式で推定した。

$$\frac{1}{h} = \frac{1}{h_{AM}} + \frac{X_5 - X_3}{\lambda_M} + \frac{1}{h_\infty}$$

- 2) 凝固潜熱……純アルミニウムの凝固潜熱は 93 Kcal/g であるが、アルミニウム合金では組成により明確でない。90 Kcal/g 附近を採用する例^{[6][7]}が多いが、この数値では温度回復が大きすぎる傾向があり、測定冷却曲線と大きくはずれる。CAST では凝固潜熱の扱いを温度回復法^{[8][9]}によっている。凝固進行中の要素では図4のように潜熱を考慮しない解析(ステップII)を行い、温度回復法によって温度回復(ステップI)をさせる。

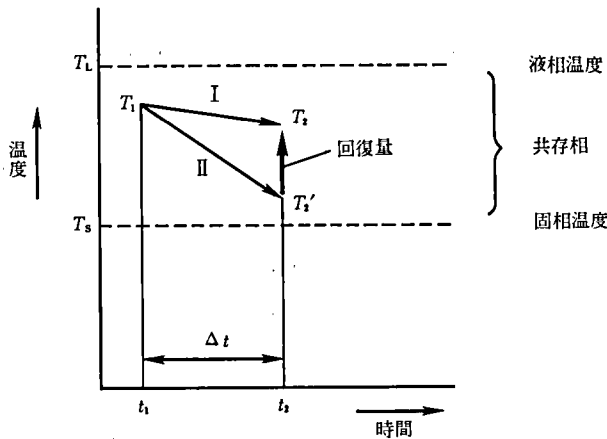


図4 温度回復のステップ
Fig. 4 Recovery of temperature

ステップIの全熱放出量は、

$$\begin{aligned} Q_I &= \int_{t_1}^{t_2} \rho \cdot c \frac{\partial T}{\partial t} dt + \Delta Q_s \\ &= \int_{T_1}^{T_2} \rho \cdot c dT - \int_{T_1}^{T_2} \rho \cdot L \frac{\partial f_s}{\partial T} dT \end{aligned}$$

ステップIIの場合は、

$$Q_{II} = \int_{t_1}^{t_2} \rho \cdot c \frac{\partial T}{\partial t} dt = \int_{T_1}^{T_2} \rho \cdot c \cdot dT$$

フーリエの保存法則により $Q_I = Q_{II}$ であり、

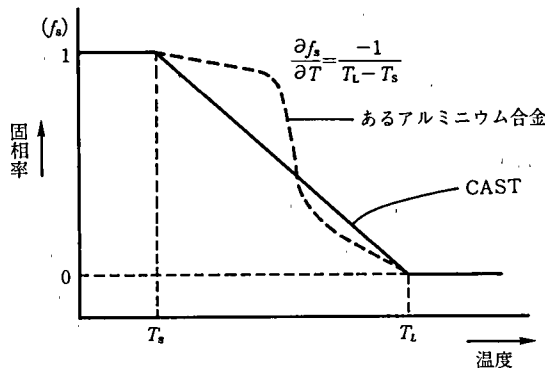


図 5 固相率と温度の関係

Fig.5 Relation of solidity and temperature

$$\int_{T_1}^{T_2} \rho \cdot c \, dT - \int_{T_1}^{T_2} \rho \cdot L \frac{\partial f_s}{\partial T} \, dT = \int_{T_1}^{T_2} \rho \cdot c \, dT$$

上式が $T_2 \rightarrow T_2$ への回復を示すものであるが、凝固率 f_s と温度との関係 $\partial f_s / \partial T$ を知ることが重要な問題となる。

この関係を正確に把握することは困難であり、CAST では線形温度回復を仮定している。

図 5 は、アルミニウム合金の固相率曲線が線形仮定に基づいていないことを示す例であるが、部分線形法（下式）の導入は計算時間の増大を考えると簡単に導入できない。

$$\int_{T_1}^{T_2} \rho_1 c_1 \, dT - \left[\int_{T_1}^{T_{m1}} \rho \cdot L \left[\frac{\partial f_s}{\partial T} \right]_1 \, dT + \int_{T_{m1}}^{T_{m2}} \rho L \left[\frac{\partial f_s}{\partial T} \right]_2 \, dT + \dots + \int_{T_{mi}}^{T_2} \rho L \left[\frac{\partial f_s}{\partial T} \right]_0 \, dT \right] = \int_{T_1}^{T_2} \rho \cdot c \, dT$$

今回は、温度に依存する潜熱を与えることにより測定冷却曲線へ近づけた。

- 3) 初期温度……現状の凝固解析の仮定として、注湯時の熱移動を考慮せず、鑄型キャビティが溶湯で満たされた時を解析開始時刻としている。キャビティ・ボリュームが大きいと、この仮定に無理があり、注湯途中にもキャビティ下部の冷却は進む。モデル 1 では、多少のこの効果の反映をねらって、初期温度の段階的差を入れている。

3.3.2 モデルの成立について

- 1) 凝固収縮流……凝固時に生じる凝固収縮や液相の流動はマクロ偏析や巣の原因となるため、CAST には凝固収縮流の処理を含む。

しかし、以下の理由で今回のモデルでは凝固収縮流を評価の対象からはずしている。

- ① 流動現象を熱現象と連動させることによる計算時間の増大
- ② モデルが小さいため、その影響は比較的小さい
- ③ 透過率などパラメタ設定が困難

- 2) 注湯時の流れ現象……鑄物の不良は湯流れに基因するものが多いため^[5]、湯流れ現象予測の必要が求められている。現行の凝固解析は湯の充填時を解析始点と

しているため、充填過程は無視される。

一方、プラスチック成形時の射出流動現象は広くシミュレーションの対象となっており、その解析ツールを利用した注湯過程解析の報告⁵⁾もある。ただしプラスチック射出成形品は基本的に薄板構造であり、流れ現象も2次元の流れである。他方、鋳物製品は中実ブロック構造が基本であり、プラスチック流動解析ツールの転用にはモデル上限界がある。

4. 解析結果と評価

4.1 測定冷却曲線との比較

モデル1, 2, 3の計算冷却曲線と計測冷却曲線の比較を図6, 7, 8に示す。すべての要素の固相率が0.71に達した時に計算を打ち切っている。モデル1, 2で断熱系塗型の効果が顕著に表れている。

4.2 ひげ巣発生位置の予測

凝固解析によるひげ巣位置予測には多くの方法が挙げられている¹⁸⁾。CASTには等凝固時間曲線法、温度勾配法、 G/\sqrt{R} 法が採用されており、それらを総合的に評価して位置予測をすることになる。

以下にそれらの特徴を列記しておく。

- 1) 等凝固時間曲線法……鋳物材料に適した限界固相率 ($0 < f_s < 1$) を与え、その固相率に達する時間を等高線で表示する。凝固していく順序が全体としてつかめる。等高線に閉ループが生じればその内側領域は孤立しており、押湯と遮断されているため溶湯補給がなく、ひげ巣が発生する可能性がある。
- 2) 温度勾配法……凝固時、ある節点が臨界固相率に達したとき、周辺未凝固要素との間の温度勾配を計算する。ある節点が凝固するとき、隣から溶湯の補給が容易に行えるかどうかを温度勾配 G ($^{\circ}\text{C}/\text{mm}$) の大小で判断する。 G が臨界値以下

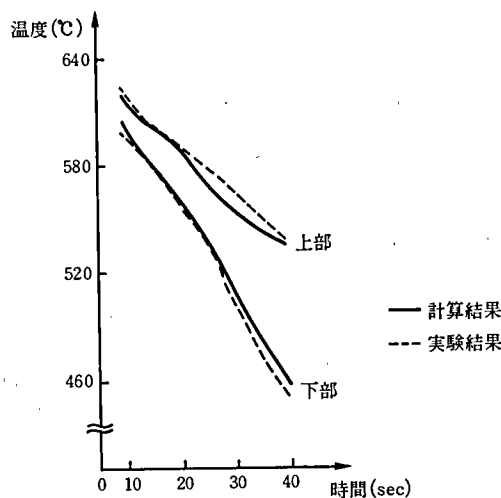


図6 冷却曲線 (モデル1)
Fig.6 Cooling curve (Model 1)

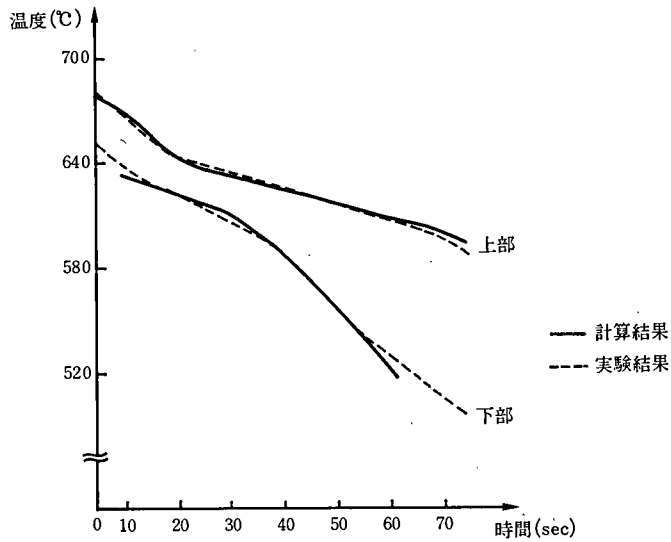


図 7 冷却曲線 (モデル 2)
Fig. 7 Cooling curve (Model 2)

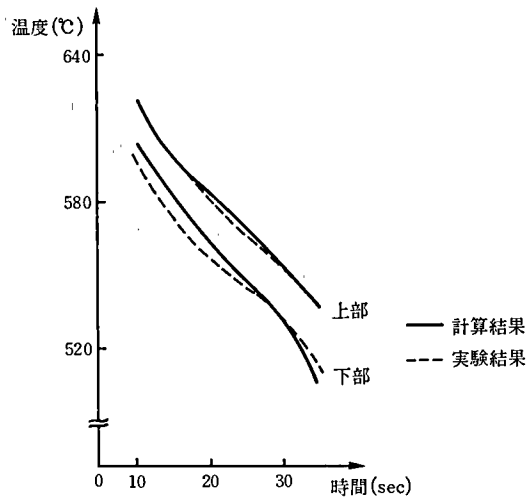


図 8 冷却曲線 (モデル 3)
Fig. 8 Cooling curve (Model 3)

であれば流動抵抗が大きく補給が困難となる。

- 3) G/\sqrt{R} 法……上の温度勾配 G と冷却速度 R を組み合わせたパラメタでその臨界値は材質や寸法に影響されないため、 G よりも評価が容易である。 G/\sqrt{R} の単位は $^{\circ}\text{C}^{1/2}\text{sec}^{1/2}\text{mm}^{-1}$ である。

4.2.1 モデル 1 (塗型無) の評価

限界固相率 0.5 および 0.7 を比較すると、等凝固時間図、 G 図、 G/\sqrt{R} 図のいずれ

も同じ傾向の等高線図が見られ、いずれを適した限界固相率 θ_L として採用するか判断できない。

等凝固時間図に関して、いずれもリングによる冷却効果のためリング間に閉ループが生じる。閉ループの大きさは $\theta_L=0.7$ の場合がより大きい面積を生じる。 $G/\sqrt{R} \leq 0.45$ 以下の領域は θ_L によらずほぼ同じ面積を生じ、この領域の実験（カラーチェック）で巣がみられる。

口絵 3 と口絵 4 は G/\sqrt{R} ($\theta_L=0.7$) とカラーチェックの比較である。ひけ巣のカラーチェック図（口絵 4）をながめると、上部よりも下部リング間に巣が多くみられ、等凝固時間の閉ループが一応の目安となることを示す。

4.2.2 モデル 2（塗型有）の評価

断熱系塗料吹付により、リングによる冷却効果が小さくなっただけ等凝固時間のリング間閉ループ領域が縮小しているのがわかる（口絵 5）。 $\theta_L=0.5$ に全要素が達する所要時間はモデル 1 で 35.6 秒、モデル 2 で 63.5 秒となっている。 $G/\sqrt{R} \leq 0.45$ の領域については逆に拡大傾向にある。

4.2.3 モデル 3 の評価

口絵 6, 7 はモデル 3 の実物図およびメッシュ分割図である。

$\theta_L=0.5$ （所要時間 29.2 秒）と 0.7（所要時間 32.6 秒）の等凝固時間図は同じ傾向を示す。円柱周辺にある多くの突起物が冷却を促進し、円柱中心に最終凝固領域が残る。

口絵 8 は $\theta_L=0.5$ での等凝固時間図である。カラーチェックの結果（口絵 9）では円柱部全体に赤点が散在するが、実物をよく見ると中心部と外周部で明らかに組織の差異が出ている。（ただし大きい赤部分は、熱電対による傷である）これは G/\sqrt{R} 図（口絵 10）の $G/\sqrt{R} \leq 0.38$ の領域に一致する。

口絵 11 はモデル上部にある大きな湯だめの効果を解析に反映させるため、最上部熱容量を 10 倍とした時の G/\sqrt{R} 図である $G/\sqrt{R} \leq 0.39$ の領域は、実験での組織の差異が出る領域とさらに良く一致する。

4.2.4 解析の総合評価

等凝固時間に閉ループが生じると、ほぼその領域が G および G/\sqrt{R} の最小値領域となる。今回の結果では、実験と照らし合わせて、 G/\sqrt{R} の臨界値は 0.38~0.45 とみられるが、その領域に相当する G の領域は 0.4~0.8°C/mm の範囲にばらつく。この理由からひけ巣判定には G/\sqrt{R} がより適している。

限界固相率は 0.5 および 0.7 を採用したが、等凝固時間図、 G 図および G/\sqrt{R} 図の上で両者の差は認められず同じ傾向を示した。計算時間短縮の意味で 0.5 がより適切であると言える。

5. おわりに

アルミニウム合金鋳物の凝固過程を解析し、実験との比較により、プログラムの評価をした。定数設定、とくに熱伝達係数の設定がむずかしいと言われているが、予想を超えるものであった。

実験に近い冷却曲線が得られれば、ひけ巣位置判定の諸手法は信頼性が高いと言える。計算効率向上、線形温度回復法の見直しなどの課題¹⁰⁾、今回除いた凝固収縮流の評価を含めて作業を継続する。

本評価作業にあたり、型製作、鋳造作業、温度計測などで光生アルミニウム工業株式会社の常務取締役 神谷俊吉氏、技術部生産技術課 課長 富田勝之氏に多大な協力をしていただいた。また、APPEX/PRE による解析モデルの作成と APPEX/POST による結果の出力に関し、ユニ・ソフトエンジニアリング(株)の渡辺美樹さんに協力いただいたことに感謝の意を表する。

- 参考文献
- [1] 大中逸雄他,「液相の流動を考慮した鋼塊の凝固解析」,鉄と鋼,67,1980.
 - [2] 大塚幸男,「自動車鋳造品での応用」,シンポジウム・コンピュータによる鋳造品の凝固解析と鋳造方案への適用,総合鋳物センター,1984.
 - [3] 西田義則,「凝固解析の可能性と問題点」,非鉄鋳物研究部会資料,1988.
 - [4] 荘加勝利,「アルミ部品への凝固シミュレーションの適用」,非鉄鋳物研究会資料,1988.
 - [5] 大塚幸男,「コンピュータによる鋳造品の凝固解析システム SOLAN の実用化と解析例」,名古屋工業試験所,非鉄鋳物研究会,資料(ii)
 - [6] 中川他,「有限要素法による凝固シミュレーション・システムの開発—そのアルミニウム合金鋳物への適用」,神戸製鋼技報,Vol.35, No.2, 1985.
 - [7] 瀬戸辰郎他,「アルミニウムダイカスト割れの解析と対策」,
 - [8] 大中逸雄,コンピュータ伝熱・凝固解析入門,丸善,1985.
 - [9] 梅田高照,「鋳造品凝固解析の基礎」,シンポジウム・コンピュータによる鋳造品の凝固解析と鋳造方案への適用,1984.
 - [10] W. Rolph, K. Bathe, "An Efficient Algorithm for Analysis of Nonlinear Heat Transfer with Phase Change", Int. J. Numer. M. Eng, Vol. 18, 1982.

執筆者紹介 松林 幸枝 (Yoshie Matsubayashi)

昭和55年 University of Witwatersrand 卒業。同年日本ユニシス(株)入社。応用ソフトウェア関連プログラムのサポート,開発業務を担当。現在,CAE システム部技術計算一課に所属。



OSIの現状と動向 The Current Status and Trends of OSI

當 麻 悦 三
E. Tohma

1. はじめに

1988年は、OSI (Open Systems Interconnection)が本格的に実装された年として、OSI 元年とも呼ばれている。

基本標準の検討が機能標準の検討に進み、実際のプロダクトに具現化され、実装に関する詳細項目の設定がなされている。この詳細項目はISP (International Standardized Profile)として、各種のパラメタの組み合わせをパターン化することにより、相互接続性の確保を容易にしようとする試みである。

本稿では技術動向として標準化が進められている内容および関連する技術の動向について述べる。

2. OSI 参照モデル

OSIの各層にそって話を進める前に、下位3層に関するISO/SC 6のテーマにふれる。

新しい基本標準のテーマとして、物理層、データリンク層、ネットワーク層に関係するものにPrivate Network (私設網)と呼ばれるものがある。このテーマが具体的に何を意味するかは詳細な検討を待たねばならないが、一言でいうならばPBX (Private Branch exchange)、およびHSTDM (High Speed Time Division Multiplexer)等を含む私設網の検討である。

これらに関するものとしてTDM (Time Division Multiplexer)、S-TDM (Statistical Time Division Multiplexer)さらにパケット交換機のNP (Network Processor)が含まれると考えられる。この検討においては、私設網と公衆網がどのように接続されるかも検討項目となるであろうし、すでに実用化されているLAN (Local Area Network)との関係も明確にされるであろう。

次にOSI環境(OSI Environment: OSIE)を越える範囲について少し述べておく。OSIの環境が

整いつつある現在、基本標準の検討はユーザにこのOSIEをいかに使いやすくするか、また分散処理環境として各業務処理が一つのOS (Open System)で処理されるのみでなく、複数のOSにまたがって統一された業務処理として実行できることの検討である。この検討のテーマはODP (Open Distributed Processing)と呼ばれ、検討が開始されたところである。

また、ユーザ・インタフェースについて、AP (Application Program)インタフェースとMMI (Man Machine Interface)の検討がある。

APインタフェースの検討は、IAP (Interface for Application Portability)のテーマにより、OSIEと業務処理プログラムのインタフェースだけでなく、OS (Operating System)と業務処理プログラムのインタフェースを含む業務処理プログラムの環境すべての検討を行って、業務処理プログラムが一つの計算機から他の計算機へ移植できるようにする検討も開始されている。

またMMIについては、ウィンドウ等の概念から使用するアイコンの統一を含めての検討が行われている。

業務手順の内容にまで検討が及ぶテーマにEDI (Electronic Data Interchange)がある。日本ではビジネス・プロトコルという用語が少しずつ一般化されているが、このEDIはまさに業務処理内容のとりきめを定めようとするものである。基本標準の検討は概念モデル (Conceptual Model)を設定し、このモデルをベースとするサービスとプロトコルを定め、各種の業務の形式を統一的に考えようとするものである。さらにEDIは一步進んで業務処理ごとに処理方法を統一し、この内容を設定することにより、各業務ごとの処理方法をまとめることを目標としている。

このように幅広い内容がOSIを取りまく環境として存在するが、以下、各層においてどんな技術が検討されているか、また一般的にどのような技術が普及しているかを述べる。

3. 物理層

物理層のインタフェース、モデムの標準は表1、表2のとおりである。表1はDTE (Data Terminal Equipment)とDCE (Data Circuit Terminat-

ing Equipment)のインタフェースであり、一般にいうモデム・インタフェースである。また表2はモデムに関する CCITT*勧告であり、回線を経由してのモデムとモデムとの手順である。基本標準の検討は物理層のサービスの定義と ISDN (Integrated Services Digital Network)コネクタ、LAN コネクタがまとめられている。

モデムの実装技術はモデムが NCU (Network Control Unit：網制御装置)と合体して、たとえばモデムフォンと呼ばれ電話網と接続されるものがある。このモデムフォンのモデム・インタフェースは従来通りの V.24 インタフェースであるが、ダイヤル機能は V.25 bis と呼ばれるビットシリアル送受によるコマンドにてダイヤルを行う機能を持っている。

表1 DTE/DCE インタフェース標準
Table 1 DTE/DCE interface standard

種 別	CCITT勧告		ISO標準
	機能	電気	機 械
電 話 網	V.24	V.28	2110 (25ピン)
	V.35	V.35	2593 (34ピン)
デジタル データ網	V.24	V.10	4902 (37ピン)
	X.24	V.11	4903 (35ピン)

また LAN の物理層としては、“より対線”，同軸ケーブル，光ファイバ・ケーブルとの接続が行われている。ISDN とのインタフェースは物理層のサービスとして，B (情報)チャンネルと D (信号)チャンネルの2種のコネクションの提供，コネクションの確立/解放が行われている。D チャンネルは，マルチポイントのコネクションを持ち送出優先制御を行っている。

さらにモデム・インタフェースに代わるものとして，GPIB (General Purpose Interface Bus)および FDDI (Fiber Distributed Data Interface)等

表2 モデムに関する CCITT 勧告
Table 2 CCITT recommendation for modems

勧 告	伝 送 速 度	変調方式	回 線
V.21	300	F S K	交換
V.22	1,200	P S K	交換
V.22bis	2,400	P S K	交換
V.23	600/1,200	F S K	交換
V.26	2,400	P S K	専用
V.26bis	1,200/2,400	P S K	交換
V.26ter	2,400	P S K	交換
V.27	4,800	P S K	専用
V.27ter	2,400/4,800	P S K	交換
V.29	9,600	Q A M	専用
V.32	9,600	Q A M	交換

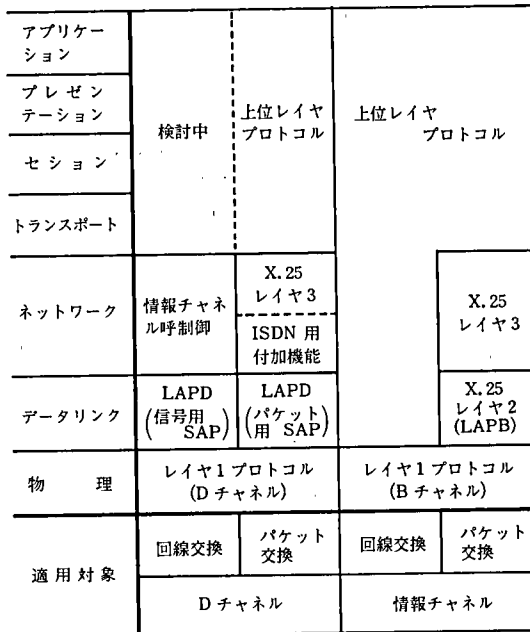


図1 ユーザ，網インタフェース・プロトコルの階層構成
Fig.1 Hierarchy structure of user network interface

*CCITT：国際電信電話諮問委員会

が用いられている。

4. データリンク層

データリンク層の検討は、マルチリンク手順がスループットおよび信頼性の向上の目的で完了している。マルチリンク手順は、複数の通信回線によって一つのデータリンクを提供するためのプロトコルである。

図2のとおり各通信回線ごとにそのリンクの伝送制御手順としてのSLP (Single Link Procedure)を持ち、これらをまとめるMLP (Multi Link Procedure)がある。個々のフレームには送信番号が付加されており、どのSLPで送信されても、受信時はMLPにおいて送出された順序と同じ順序で受信できる。また通信回線上の誤まりは

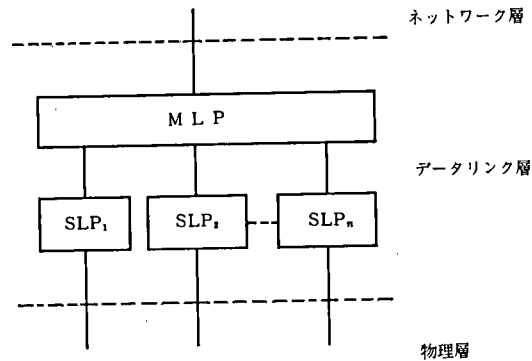


図2 MLPとSLPとの関係
Fig. 2 Relation between SLPs and an MLP

各SLPにより再送が行われ、正しく伝送される。

また、衛星通信用としての送信シーケンス番号の拡張によるモジュロ128の項目についての検討が終わっている。この拡張はフレームを応答なしに先送りできる数が127まで可能であり、遅延の大きい通信路への対応が可能となる。その後の検討として32ビットのFCS (Frame Check Sequence), コマンド/レスポンスのうちのS-REJ (Selective Reject)の使用およびXID (Exchange Identifier)の使用の検討が完了している。

LANにおいてはCSMA/CD*, トークンバス, トークンリングの手順およびLLC (Logical Link Control)がまとめられている。LLCのコマンドの例を表3に示す。

ISDNにおいてはLAPDが用いられる。LAPD (Link Access Protocol D)は制御チャンネルの手順であり、LAPBは情報チャンネルの手順である。これらはハイレベル伝送制御手順のABM (Asynchronous Balance Mode)をベースとしている。LAPDとLAPBとの相違は次の通りである。

- 1) 集中型のマルチポイント・コネクションはフレームの送達を確認できないため、LAPDではブロード・キャスト (同報)を行って送達確認は行わない。
- 2) LAPDでは、制御に対する汎用的な考え方からSABMEによるモジュロの拡張およびアドレスの拡張がある。

UNIX**の世界ではTTYの手順が多く用い

表3 論理リンク制御コマンドと応答一覧
Table 3 List of logical link commands and responses

タイプ	型式	略称	コマンド	応答	意味	情報フィールド	
Type 1	U	UI	○		非番号制情報	有	
		TEST	○	○	テスト(折り返しテスト)	有	
		XID	○	○	ID変換(LLCクラスとウィンドサイズを伝達)	有	
Type 2	I	I	○	○	順序番号付き情報	有	
		S	RR	○	○	受信可状態を表す	無
			RNR	○	○	受信不可状態を表す	無
	REJ		○	○	再送要求	無	
	U	SABME	○			拡張非同期平衡モードへ移行	無
		DISC	○			切断モードへ移行	無
		DM		○		切断モード中	無
		UA		○		非番号制コマンドに対する応答	無
			FRMR		○	PDU異常報告	有

* CSMA/CD : Carrier sense multiple Access with Collision Detection

** UNIX : AT & T ベル研究所の登録商標である。

られており、これらの手順の中にハイレベル伝送制御の良い点を導入したのとして、MNP (Microcom Networking Protocol) および Kermit がある。MNP モデムは HDLC の簡易手順をハードウェアとして取り入れたものであり、Kermit はこの簡易手順をソフトウェアにより実現している。

今後多く用いられる手順は、HDLC-ABM と NRM (Normal Response Mode) である。

5. ネットワーク層

ネットワーク層の packets 交換手順は、CCITT X.25 勧告として packets 交換網 (DCE) とホストまたは端末 (DTE) との間の手順が定められているが、この手順によると DTE と DTE が packets 交換網を経由せずに通信することが不可能であった。このための DTE 用 X.25 のサービス、プロトコルの検討が行われた。

さらに CCITT では X.25 88 年版がまとめられ、今後はどのように 88 年版 X.25 を機能標準とするかが課題となっている。

また、網と網との接続に関連してネットワーク・アドレスの検討が完了している。CO (Connection Oriented) のネットワーク・サービス、プロトコルに加え、CL (Connectionless) のサービス、プロトコルの検討が行われた。これは、たとえば LAN において接続がない CLNP (Con-

nectionless Network Protocol) としてまとめられている (図 3)。

ISDN での CO のネットワーク・サービスの特徴は以下の通りである。

- 1) 回線交換用接続と packets 交換用接続の 2 種を提供する。
- 2) 接続確立前に端末相互の通信の可能性を確認する。
- 3) 回線交換用接続ではトランスペアレント/ノントランスペアレントを選択できる。

現実の product としては、DDX packets 交換網への接続を可能とするよう実装されている。DDX packets 交換網は、現在 X.25 80 年版をベースにデータ長の拡張等が行われている。

6. トランスポート層

実装が進んでいる層であり、この層の中での主なクラス分けは次の通りである。

- 1) クラス 0 ……このクラスは多重化機能および障害回復機能を持たない最も簡易なクラスである。
- 2) クラス 2 ……このクラスは多重化機能は持つが、障害回復機能を持たない。
- 3) クラス 4 ……このクラスは多重化機能を持ち、障害回復のための機能を持つ。

これらの適用において WAN (Wide Area Net-

	[CO ネットワーク・サービス]			[CL ネットワーク・サービス]		
レイヤ 3	X.25 84年版 DTE 標準	X.25 コンバージョン プロトコル X.25 80年版 DTE 標準	X.25 84年版 DTE 標準 X.25による LAN 上での CO サービス 提供	CL 下位 サービス上 での CO サービス CL プロト コル	CL プロトコル + サブネット ワーク・サ ービスへの CL プロト コルのマッ ピング	X.25 上での CL サービス の提供方法 + X.25 ベース
レイヤ 2	WAN の標準		LAN の標準			
レイヤ 1						

図 3 ネットワーク層のプロトコル
Fig. 3 Protocol in network layer

work)ではクラス0, クラス2が用いられ, LANではクラス4が用いられる。ここでISDNについて考えてみる。ISDNでは回線交換モードとパケット交換モードがある。回線交換モードの時はどんな伝送制御手順が用いられるかを考える。例としてハイレベル伝送制御手順があり, 回線上での誤まりは十分にチェックされ, データの保証ができる。パケット交換モードは, パケット交換の手順によってデータの保証がなされる。したがって, トランスポート層としてはクラス0と2のいずれかが用いられる。

このように実装が進む中でコネクション型の検討が終了し, コネクションレス型の検討も終わっている。また, 今後の検討テーマの一つにサービス品質 (Quality of Service : QOS) がある。このQOSはトランスポート層の存在の根底となる考え方である。トランスポート層の検討とは, 下位のサービスの品質の異なるものを上位の層に均質のサービスを提供する役割についての検討である。下位層からの品質が異なるものに対し, 上位に均一のサービスを提供するためには, トランスポート層が異なる内容の機能を持たなければならない。現在はサービス品質の分類が完了しており, 上位に提供すべき品質が定まれば要求される品質との関係から要求される機能が定まるものとなる。

このような本質的な検討とともに, 形式記述技法 (Formal Description Technique) により表記する層としてトランスポート層が選ばれ, LOTOS (Language of Temporal Ordering Specification) による記述がなされている。さらに, 今後は層管理の検討がトランスポート層を対象に進むであろう。

7. セッション層, プレゼンテーション層, アプリケーション層(1)

セッション, アプリケーション, プレゼンテーション層の概念は, 上位層アーキテクチャ (Upper Layer Architecture : ULA) と呼ばれまだ未分化の内容を持っている。これらの機能分担と, それぞれの機能の明確化が継続して検討されている。

セッション層は, 下位層からの続きというよりも上位のアプリケーションからの窓口であり, 応用プロセスからの要求に合わせた会話制御, コネクションの確立と解放, データ交換のための同期制

御, アプリケーションに応じた半二重/全二重モードのサービスを提供する。実装に向けての条件として次のものがある。

- 1) カーネルは必須
- 2) ケーパビリティ機能単位はアクティビティ管理とともに用いる。
- 3) 例外報告機能単位を用いるのは, 半二重モードの時のみである。

CCITT 勧告 T. 62 との整合がとれた今, 今後の検討課題はデータ交換の中断と再開方法の明確化, 全二重モードの同期の取り方の明確化であり, アプリケーションからのニーズを十分に配慮した細やかな地道な努力が続けられている。

プレゼンテーション層は暗号化, データ圧縮, およびパススルー等が一般的な機能である。実装において, ファイル転送ではデータ圧縮が用いられる場合がある。また処理効率に関してパススルーの考え方が一部提案されている。しかし機能の分担と役割の明確化の観点からパススルーの考え方 (一つの LAYER を機能させなくする) は正しい論理の展開とは言えない。

プレゼンテーション層の大切な役割としてシンタックス (構文) の変換がある。構文変換はアプリケーション層で用いられる構文 (抽象構文) をデータ転送を行うための構文, すなわち転送構文に変換およびこの逆の変換をいう (図4)。実装においてパススルーと同じようにこの構文変換を実際の処理として実行せず, あらかじめ上位から転送構文を流し込むことが多く行われている。この処理方法は今後整えられるべきものである。

また構文変換に関連してコンテキスト制御がある。コンテキスト制御は, 各種の抽象構文を各種の転送構文に対応付けるための制御を言う。簡単な方法として, デフォルトとなっているものにバイナリの転送を行う方法がある。

また, 最近の最も大きな話題となっているものに抽象構文記法 1 (Abstract Syntax Notification 1 : ASN. 1) がある。今後の拡張が考えられるため 1 の番号が付与されている。

この機能拡張とは別の観点で ASN. 1 の中で用いるコードとして, 日本語をどのように処理させるかという問題がある。国際標準の ASN. 1 は 1 オクテットで十分に内容が表記できるため, 日本語等のマルチ・オクテットの考え方は取り入れられていない。マルチ・オクテットを用いる国は,

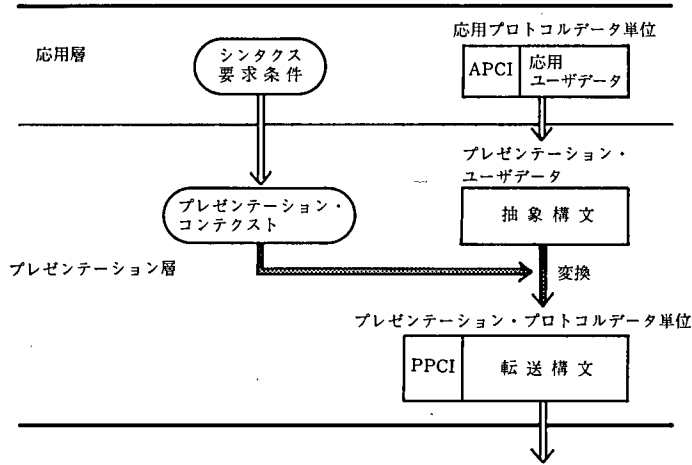


図4 構文の変換
Fig. 4 Transformation of syntax

日本、韓国、中国等わずかである。ASN. 1のmultiobjectの検討は日本が自からベースを作成し、実装についても日本のメンバが力を合わせ世界に通用するプロダクトを開発し、これを広める必要がある。

アプリケーション層内の構造もULAと同様に未分化のところがあるが、共通に用いられる機能とアプリケーション固有のものがある。

ここでは、そのうち共通に用いられる要素のいくつかについて述べる。この代表がACSE (Association Control Service Element)である。

主な機能として次のものがあり、詳細の検討が行われている。

- 1) アソシエーションの確立・解放
- 2) 応用コンテキスト名とそのパラメータの交換
- 3) データ転送とプレゼンテーションの機能を用いた会話制御

またCCR (Commitment Concurrency Recovery)は、分散処理環境における共通機能である。これらは他の固有アプリケーションとの関連が深く、たとえばトランザクション処理はCCRを用いている等、相互に関連する機能の検討が行われている。これと同様にROS (Remote Operation Service)があり、遠隔地の操作に対して制御、管理する機能の検討も行われている。

8. アプリケーション層(2)

アプリケーション層の固有のアプリケーション

に関するものについて述べる。固有アプリケーションのFTAM (File Transfer, Access and Management), MOTIS (Message Oriented Text Interchange Service), RDA (Remote Database Access), ODA (Office Document Architecture)についてはそれぞれファイル転送、電子メール、データベース・アクセス、ドキュメント構造というテーマでの実装が行われている。

ユーザの使用についてもFTAMの利用検討が始まっている。FTAMに今後期待されることは機能の充実である。現在はファイルの構造もフラット型と呼ばれ、階層構造を持たない。転送時の誤まりに対する回復も、途中からの再送ではなくファイル全体の再送である。ファイルへのアクセスおよびファイル管理の実装については今後検討され、充実されるものである。FTAMは応用層の共通機能とは明確に機能分担がなされている。このプロトコルが、現在多く用いられている全手順、J手順のファイル転送に取って変わる日が真のFTAMの利用の時である。

MOTISは、本来84年版CCITT勧告のMHS (Message Handling System)から出発しており、その機能は拡張されMHSと差異を持つものとして存在した。しかしこの差異については、ISOとCCITTのJoint Meetingにおいて検討が終了し88年版での差異はなくなった。電子メール・システムの利用は米国、欧州でかなり幅広く利用されており、日本においても実装の段階から利用の段階へと進展しつつある。日本人が用いる電子メー

ルは何語ですか。こんな低いレベルの質問に対し当然「日本語です」との返事が返ってくる。では日本人が米国の人と電子メールを行う時の用語は何ですか。これも簡単に「英語です」との返事が返ってくる。この二つから考えられることは、日本の電子メールは世界の基本標準の電子メールと差異があるということである。このための実装は日本の MHS としてのプロダクトとなる。88 年版の MHS の機能は 84 年版の MHS の機能に比べ、いくつかの拡張機能がある。

88 年版の MHS が現在実装されている MHS と、どのような関係にあるかが今後のポイントとなる。一つの考え方として 88 年版 MHS は現在実装されている MHS と接続でき、メール機能を有効に使用できるだけでなく、それ自体は拡大された機能を持っているといえる。この新しい機能のリリースをどのような形式で、いつ行えばよいかの検討が必要となる。電子メールのプロトコルは、P 1 および P 2 プロトコルの統一は完了している。個々の端末/ワークステーションとの間の、たとえば P 3 プロトコルまたは P 3+ プロトコルの実装の統一が今後検討され、この実装が完了した時が MOTIS の完成期といえる (図 5)。

ユーザが最も期待している固有アプリケーションに TP (Transaction Processing) がある。この基本標準の検討はやや遅れ気味であるが、実装に向けての検討が開始されている。

さらに実装の検討が進んでいるものにディレクトリ (The Directory) がある。この実装の検討は第 1 段階が終了し、最終版の基本標準をベースに実装内容の更新が行われている。しかしこのテーマも問題を持っていないわけではなく、ディレク

トリの真のアプリケーションが何かということが問われている。電話番号の案内は“人間が受け、人間が答える問い合わせシステム”により融通性のある、柔軟なシステムとなっているが、計算機システムによる問い合わせでは、問い合わせの方法もかなり繁雑となり、そのシステムの内容を十分知っている人のみが利用できるという限定されたものにもなりかねない。この問題に対する回答の一つとして、MOTIS システムと組み合わせて使用し、メールの宛先等の案内に利用するというのがある。これも限定された利用の一つであるが、このような段階を経て、さらに一般的な利用への拡大がなされるであろう。

次に実装の計画がなされているものが OSI 管理である。OSI 管理は何をするのかという問いに対して、管理フレームワークがその解答を与えている。管理のサービスは何か、また管理の対象は何か、ということが現在検討されている。管理はその目的によってその形が変わるため、次のような各種の管理が考えられる。

- 1) 障害管理
- 2) 構成管理
- 3) 性能管理
- 4) 課金管理
- 5) 機密管理

いままでは、これらの管理を機能単位とする分類に基づいて検討されてきたが、今後はこれらの枠をはずし機能を単位とする分類に変更された。その中で一つの機能が何を対象として、何をサービスするかの検討が行われることになった。

JTM (Job Transfer and Manipulation) すなわちジョブ転送については OS に依存するところが

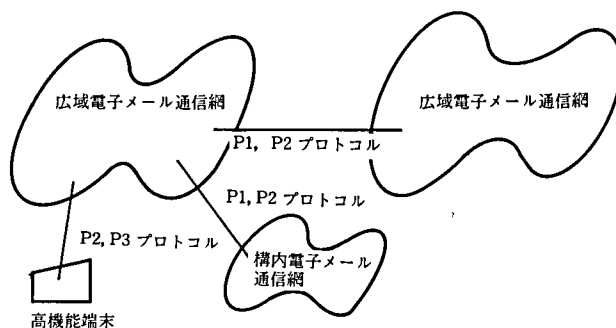


図 5 MHS のプロトコル
Fig. 5 MHS protocol

多く、各メーカーが提供する RJE (Remote Job Entry) で十分との考えが強い。

またテレマティックサービスとして、次に示す機能の実装が行われている。

- 1) G4ファクシミリ
- 2) ビデオテックス
- 3) パソコン通信
- 4) テレテックス
- 5) ミクストモード

これらの発展は、OSI をベースとしながら進むという形態で行われている。

G4ファクシミリは機能標準が定まりプロダクトが存在し、相互接続テストも完了して、各社の商品化がなされている。またこれに関して、NTT がサービスするファクシミリ通信網のセンタと、外部コンピュータの接続プロトコルも OSI をベースとするものに更新されることが決まっている。

ビデオテックスも同様であり、現在のビデオテックス網のセンタと外部コンピュータとの接続は、古いバージョンではあるが OSI のプロトコルがすでに用いられている。

パソコン通信も一部 OSI の概念が取り入れられており、OSI をベースとした実装が行われている。

テレテックス、ミクストモードの用語は最近は

あまり用いられないが、文書構造の検討においてテキスト形式、これ以外のものを統一して処理する体系があり ODIF (Office Document Interchange Format) 等の検討が進んでいる。

最後に仮想端末の実装に向けて検討が開始された。VT (Virtual Terminal) の実装の方向付けはまだ明確ではないが、ここ一年で大きな進展を示し、実用化が早まることを述べておく。

9. おわりに

OSI の技術ということでもまとめを行ったが、技術的な詳細にふれ得なかったこと、今まで述べたこと以外にも、たとえば次のテーマがある。

- 1) コンフォーマンス試験
- 2) セキュリティ
- 3) 名前とアドレス
- 4) 登録機関

これらの検討が今後重要となってくる。

日本ユニシス (NUL) では各種の検討に参画し、さらに広い分野での活躍が期待されている。

なお、本稿の作成にあたり、御支援頂いた方々に感謝の意を表する。

参考文献

- [1] 情報処理, 情報処理学会, Vol. 26, No. 4, pp. 289~436.
- [2] 情報処理, 情報処理学会, Vol. 26, No. 4, pp. 662~724.

IEEE ソフトウェア編集委員会編

IEEE ソフトウェア'88

岩波書店, A4判, vi+186pp., 1988, 3,500円

便利な書籍が出版されたものである。本書が出版されたいきさつから紹介しよう。出版を企画した人の意図は以下のものである。

ソフトウェア生産は古くから伝統のある産業とは異なり、前近代的な生産形態をとっているが、その原因の一つにソフトウェア生産技術の、研究部門から生産現場への技術移転がうまくいっていないことがある。近年、研究室ではいろいろな研究成果が生み出されているが、現場の技術者は日常の業務に忙殺されて研究成果に目を配る時間的余裕がない。

この問題に対処するために、IEEE の Computer Society では 1984 年から研究室の成果を企業の中堅技術者や他分野の研究者に理解でき、かつ現場で適用できるように噛み砕いた形の雑誌“IEEE Software”を発刊している。隔月刊の雑誌であるが、日本ではこれを翻訳して年 1 回の単行本として出版する計画であり、本書はその最初の版である。

この第 1 号は、創刊号から 1987 年 9 月号までの計 20 号、70 論文の中から 13 編を採用している。こうした事情から掲載されている論文は粒揃いであり、読み応えがある。

目次は以下のようになっている。

- ① プログラミング—魔術か科学か? C. A. R. Hoare
- ② Cedar への旅 W. Teitelman
- ③ 視覚言語とは S. Chang
- ④ プログラミング言語と抽象化 M. Shaw
- ⑤ Smalltalk によるプロトタイピング J. Diederich
J. Milton
- ⑥ プログラミングの進化論 B. Cox
- ⑦ 構造化プログラミング—回想と展望 H. Mills
- ⑧ ソフトウェア再利用は役に立つか? T. Biggerstaff
C. Richter

⑨ ソフトウェア部品の分類と再利用

R. Prieto-Díaz

P. Freeman

⑩ 部品からのソフトウェア合成 G. Kaiser

D. Garlan

⑪ 仕様をきちんと書くには B. Meyer

⑫ 計算機システムを設計する人のために

B. Lampson

⑬ [ソフトウェア評論]

デスクトップ・パブリッシング

G. Gruman

目次から解かるように、扱っている対象が多岐に渡っているため、読者がどの論文を面白いと感じるかは読者の関心によって異なるだろう。以降の紹介も紹介者の関心に基づくものであって、論文ごとに公平な、と言うよりも独断的なものなることを最初にお断りしておきたい。

さて、本論に入ろう。C. A. R. Hoare の論文(①)は、同氏のオックスフォード大学計算学教授就任記念講演(近く「プログラミングの数学」という題で雑誌“bit”から邦訳が出ると聞く)の理論的基調に基づいている。記念講演が、プログラミングは数学的営為に基づくものであることを理論的に実証しているのに対して、論文では理論的に深入りするのを避け、数学的証明という武器によって他の工学分野の成果物と同じように信頼性の高い、プログラムという成果物をつくることを主張している。他の職業と比較して言う。「職業としてプログラミング業務を行うためには、基礎となる数学理論に基づき、すでに確立された他の工学分野の先例にならなければならない。これは教育を改善することによって実現することができる」さらに、「われわれはごく最近になるまで、コンピュータ・プログラミングの数学的かつ論理的な基礎を確立することができなかった。しかし今やわれわれは、技師が橋や道路の位置を測定するのと同じ正確さでプログラムの仕様を作り上げることができ、技師がこれをもとにして橋が落ちることがないことを保証できるのと同じ確かさで、仕様に合致していることを保証したプログラムを作成することができる」このような技術はいくつかの研究によって裏付けられていると言

い、われわれになじみのあるいくつかの論文を紹介する^{[1][2][3][4]}。

数学的証明などと言うと、研究室内での小さな実験と思われがちであるが、大規模プログラミング・プロジェクトに適用可能な時代になっているとも言う。記念講演では、数学的方法を実務規模の問題に適用できるようにするためのいくつかの試みがオックスフォード内で行われ、成果を上げつつあることを報告している。ここでは、「……私は、形式的数学的手法をスケールアップして、工学的な応用に対して適用する時がきたと信じている。これは大学あるいは工芸学校と、産業界あるいはソフトウェアハウスとの共同開発プロジェクトという形でもっとうまく達成できると思う」という。

形式的数学的方法による実現に一抹の疑問を抱えながら、それとは無縁な日常業務の間で時間をやりくりしながらそれを勉強している身にとっては、C. A. R. Hoareのご託宣は百万の味方を得た思いがする。それにもかかわらず、わが身のまわりを見廻すと、本当かなという思いは禁じ得ない。こうした思いに実証的に反駁してくれるのが論文⑦である。

著者の H. Mills は、構造化プログラミングが話題になっていた頃、それを産業界に適用しチームプログラマ・チームなどの概念形成を助け、E. W. Dijkstra とは異なった側面で構造化プログラミングの発展に貢献した人として知られている。Mills は構造化プログラミングの活動を回想し、Dijkstra の最初の議論はプログラムの制御構造を単純化して、プログラムの正当性の証明を短くすることに中心を置いていたことを思い出させる。「Dijkstra の最初の構造化プログラミングに関する論文では、文法、表記法、読みやすさ、段階的詳細化あるいはトップダウン開発と言ったものは述べられていない。むしろ彼の構造化プログラミングの主な議論は、プログラムの正当性の数学的検証を短くするということであつた。それは不思議な議論のように思えた。というのは当時ほとんど誰一人として自分のプログラムが正しいことを証明しようなどと悩む者はいなかったからである。しかし、それは現在もまだ展開されつつある、示唆に富んだ予見であつた。」こうして Mills は、Dijkstra の初期の精神を今産業界で実現しようとする。ただ彼は Hoare の公理的検証に代わつ

て関数的検証という別の形式を用いる。産業界に席を置く人らしく、その他に、配列を使わないデータ構造化プログラミング、単体デバッグなしの統計的テストなどの工夫も採り入れている。こうしたやり方で数万ステップ規模のプログラムを開発し、従来に比して飛躍的に改善された例を報告している。関数的検証や統計的テストなど、この論文からだけではもうひとつ解りづらい所もあるが、なんと言っても実務に適用し高い成果を上げている点が強みである。

第3にお薦めの論文は⑩である。この論文はこれまで紹介した①や⑦のような衝撃はない。この種の論文としてとくに目新しいものでもないが、形式的仕様記述のチュートリアルになっている。本論文のねらいとして、「自然言語仕様記述法を批判し、その後数学的手法について述べる。これにより、要求仕様記述における自然言語の弱さと形式性の強さを示す」と言っている。Naur と Goodenough/Gerhart の仕様を例に論を進める。紹介者には、以前これらの仕様になじみがあったから、とくに親しみを感じたのかもかもしれない。著者は、自然言語の持つ理解しやすさという長所まで否定しているのではなく、形式的仕様記述を行うという回り道をとることによって自然言語の持つ曖昧さや冗長さを取り除くことができ、自然言語表現の改善に役立つと主張している。今話題のシグマシステムのマニュアルを執筆した人には是非読んでほしいものである。次年度に文書化の専門家、つまり計算機の実験家、つまり計算機の実験家に執筆させ直す計画があると聞くが、方向が違ってないだろうか。急がば廻れである。

つぎに論文④を紹介する。ここでは抽象データ型の歴史的回顾を行い、抽象データ型がプログラミングに有効な武器になっていることを示す。小さな例題を FORTRAN, Pascal, Ada で実現して見せその有効性を説く。われわれにもいよいよ Ada が使える時代になった。楽しみが一つ増えた。

ソフトウェア部品の再利用に関する話題は三つ(⑧, ⑨, ⑩)あるが、これらはこの分野の理論が依然として発展途上にあることを示している。

Small talk にまつわる話題が二つある(⑤, ⑥)。

日本語で書かれているからといって簡単に読めるとは限らない。かなり骨が折れるものもある。

その中で気楽に読めるものもある。「ためになるアドバイスと逸話を集め、…経験豊かな設計者たちの知恵を授ける」論文⑩である。こうしたサービス精神があるのもうれしい。

Xerox PARC でつくられた Cedar プログラミング環境（高品質グラフィックス、洗練されたエディタとドキュメント支援機能、プログラムの作成やデバッグにおいてプログラマの使う各種のツール、などを組み合わせた最新のプログラミング・システム）を紹介した論文⑫を読むと、毎度のことながら、どうして日本ではこうしたセンスのあるシステムが生まれにくいのかと思う。マウスとか、アイコンとか、Unix*とか、外国で生まれた画期的なアイデアは、出来ばえはともかく数年を経ていつのまにか日本に定着している。この傾向は今後も続くのだろうか、と論文の中味よりもついそんなことを考えてしまう。

程度の差はあれ、われわれ企業内中堅技術者には語学上のハンディキャップは大きい。この種の論文が1,2年の遅れで日本語で読めることは大変ありがたい。今回は多くの論文の中から厳選しているから粒がそろっているのかもしれないが、質を落とさず毎年継続して出版されることを希望する。スポーツ観戦によって得られる感動は翌日になると消えてしまうが、読書から得る感動はもう少し長持ちするからいい。

*UNIX は、AT & T ベル研究所の登録商標である。

参考文献

- [1] O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Structured Programming, Academic Press, New York, 1972.
- [2] N. Wirth, Systematic Programming, Prentice-Hall, Englewood Cliffs, N. J., 1973.
- [3] M. A. Jackson, Principles of Program Design, Academic Press, New York, 1975.
- [4] E. W. Dijkstra, A Discipline of Programming, Prentice-Hall, Englewood Cliffs, N. J., 1976.

(ソフトウェア生産技術一部 峰尾欽二)

☆

J. W. Lloyd 著

佐藤雅彦, 森下真一 訳

論理プログラミングの基礎

産業図書, A 5 判, vi+149 pp., 1987, 1800 円

論理プログラミングは1970年代初頭に誕生し、成長している分野である。論理プログラミング分野の発展は、日本の第5世代コンピュータ・プロジェクトにより大きく影響されている。昭和63年11月28日から12月2日にかけて東京でFGCS'88が開催され、国内外より多くの研究者が参加し、論理プログラミングの研究成果が発表された。中心は並列論理プログラミングであり、第5世代コンピュータ・プロジェクトの中期の成果である並列推論マシンのプロトタイプがデモされていた。

本書は、論理プログラミングの数学的基礎付けに関する重要な仕事を体系的にまとめあげた最初の解説書である。今まで、個々の研究論文を広く探し求めなければ得られなかった基本となる理論的成果を、統合的かつ網羅的に整理することを目指して著されている。本書の紹介に入る前に、論理プログラミングの背景、考え方を述べる。以降の記述は第1節よりの抜粋である。

論理プログラミングは、1970年代の初頭に始まったが、それは自動定理証明と人工知能の初期の仕事が直接的にもたらしたものである。自動演繹システムを構築することは、人工知能を生み出す目標の中心に位置している。1972年KowalskiとColmerauerは、「論理はプログラミング言語として使うことができる」という基本的な考え方に到達し、PROgramming in LOGicの頭文字を取ってPROLOGという言語を作った。今までの論理プログラミングの研究で最も重要かつ実用上の成果の一つは、この言語PROLOGであり、それは論理の一部分であるホーン節に基づいている。ホーン節に制限し、証明戦略を制限したことにより、多くの論理プログラムはほとんど決定的に動き(言い換えると、ほとんど後戻りせずに動く)、そのことが演繹(deduction)を計算と考える有用性の根拠となっている。

Kowalskiによる、論理プログラミングの中心的考え方の一つは、アルゴリズムは論理と制御の二つの異なる部分からなるということである。論理とは解くべき問題は何か(what)であるかについて

ての記述である。一方制御とはそれをどう (how) 解くかについての記述である。論理プログラミングが理想とするのは、プログラマはアルゴリズムの論理の部分のみを指定しさえすればよく、制御は論理プログラミングのシステムが勝手に実行してくれるような状況である。残念ながらこの理想は、現在の論理プログラミングのシステムでは、まだ成し遂げられていない。

論理プログラミングの理想を実現するには、解決しなければならない二つの大きな問題がある。その一つは制御の問題である。現状では、プログラマは数多くの制御の情報を与える必要があり、一部は節と節内の原始論理式を順序づけることで行われ、一部は論理以外の制御機能、たとえばカットを使って行われる。もう一つの問題として、否定の問題がある。ホーン節は十分な表現力を持っていない。それゆえ PROLOG システムでは節の本体に否定のリテラルを書くことを許しているが、否定そのものを実現しているのではなく、失敗を否定とみなす規則 (negation as failure rule) という問題の多い不完全な形の規則を実現しているだけである。

論理には他の二つの解釈がある。一つはデータベースとしての解釈である。この場合、論理プログラムはデータベースとみなされる。結果として、関係型データベースをととも自然にかつ強力なものへと一般化でき、そこでは関係型データベースは、基礎単位節のみから構成される論理プログラムに対応する。このようにデータ、プログラム、問い合わせ、ビュー、安全性の制約を統一的に扱う言語として論理を使うという発想には、理論的にも実用上も大きな潜在的可能性がある。

論理のもう一つの解釈として、プロセスとしての解釈がある。この解釈では、ゴール $\leftarrow B_1, \dots, B_n$ を並列に動くプロセスの系と考える。計算の1ステップは、一つのプロセスをプロセスの系 (プロセスの呼び出しに適合した節の本体に現れるプロセス) に変形することである。共有される変数は、プロセス間の通信チャンネルとして働く。プロセスとしての解釈を基礎にした並列型 PROLOG が、現在いくつかある。このような解釈により、論理はオペレーティング・システムへ応用することや、オブジェクト指向プログラミングに使用することが可能である。

このように論理が、計算機科学において外見上

は種々異なって見える側面に対し、一つの形式的枠組みを与えている。論理は一般目的用の問題解決言語と、オペレーティング・システムの記述に適した並列型言語を与え、データベース・システムの基礎を与える。この応用範囲の広さと、論理プログラミングの持つ簡潔さ、洗練さ、統一的な効果とが合わさり、重要かつ影響力のある将来が約束されている。また論理的推論は、計算の基本的単位になろうとしている。

本書は、4章19節より構成されている。読者には予備知識として、PROLOG等の論理的言語についての多少の知識、およびある程度の数学的素養、たとえば数学的帰納法の議論や論理記号による表現に慣れていることを仮定している。

第1章は、論理プログラミングの宣言的意味論を紹介している。まず論理プログラミングの簡単な導入をした後に、論理プログラムの基本的な記号と用語を解説し、論理プログラムの解釈とモデルについて述べている。これらが宣言的意味を与える。論理プログラミングのシステムの主な目的は変数の束縛を計算することであるので、正解代入 (correct answer substitution) の概念を説明する。正解代入は、プログラムとゴールから得られる意図した出力に対する、宣言的理解の仕方を与えてくれる。また単一化子 (unifier) と単一化アルゴリズム (unification algorithm) について詳しく議論されている。その後、不動点についての基本的な結果を証明し、最小エルブランモデルがプログラムの論理的帰結となる基礎原始式の集合と、正確に一致することを示している。そして、最小エルブランモデルの不動点による重要な特徴づけを示している。

第2章は、論理プログラミングの手続き的意味論を扱っている。手続き的に考えたとき、正解代入に相当するのは、計算解代入と呼ばれるものであり、それはSLD-導出を使って定義される。この章では、すべての正解代入は計算解代入の代入例になることを示す。これによりSLD-導出の健全性と完全性を立証する。他に証明されている重要な結果は、計算規則の独立性である。また、PROLOGの実現に関する二つの実用上の側面についても議論されている。それは単一化アルゴリズムから出現チェックがはずされていることと、制御機構カットである。カットはプログラムの宣言的意味に影響はしないものの、反駁手続きに望

ましくない不完全性の影響を落とす可能性があることを述べている。

第3章は、否定のさまざまな形態について述べる。肯定情報のみがプログラムの論理的帰結であるために、否定的な情報を演繹するには特別な規則が必要である。こういった規則の中で最も重要なものは、閉世界仮説と失敗を否定とみなす規則の健全性と完全性を示している。

第4章は、無限に続くプロセスの意味を解説している。並列プログラムの書ける PROLOG の出現により、このような意味論は理論的にも大変重

要になっている。

なお、原書においては記号がすべてローマン体で表現されているが、訳では数学書における慣習に従い、定義や定理中で使われる抽象的な議論をする際の記号をイタリック体で表し、具体例中の記号をローマン体で表している。翻訳もこなれた日本語であり、読みやすいものになっている。

本書は、論理プログラミングの数学的基礎の解説であるので、論理プログラミングの理論的背景を理解したい読者にとって一読に値する解説書である。

(知識システム部 森澤好臣)

ユニバック技報 第11号～第16号

次のバックナンバーのうち、*印のあるものは多少残部がございます。

ご希望の方は、弊社技術情報サービス部までご連絡下さい。

第11号 (1986年8月)*

- 論文 2相流解析コードの開発.....二ノ方寿, 岡野豊明
報告 複数の曲面を接続するフィレット曲面創成法橋本可輝
UNIVAC シリーズ 1100 用 Ada コンパイラ・システムG. Snyder, D. Wallace
Byron 1100 プログラミング支援環境M. Gordon, H. Turkle
M. Larsen, D. Ortmeier
JSD 仕様の実行系の試作加藤潤三
ストリームを用いた論理型言語インタプリタ大田一久
バックパネルの潜在的不良検出手法小塩英造, 中條幸雄, 三位 潔

TECHNOLOGY TREND

- 専門家システム構築ツール KEE橋本和博, 佐藤公一
事務文書体系とメッセージ指向文章交換系若鳥陸夫
インテリジェントビルの動向大嶋重光

第12号 (1987年2月)*

- 論文 ボトルネック型ブロック割当て問題前田英次郎, 森本皓夫, 保科 剛
航空運賃割引管理エキスパート・システム.....R. Wilson
SCULPTOR における境界切削工具経路の計算方法酒井喜嗣
広域ネットワークにおける高水準 NMS の開発井上 隆
光ディスク媒体の信頼性試験に関する考察大石完一
UNIVAC 2200/200 シリーズのハードウェア技術蛸子斌博

TECHNOLOGY TREND

- 文章推敲支援システム宮本義昭
入出力インタフェースの標準化動向森 宗正

第13号 (1987年5月)

- 論文 静電気スパーク放電による電磁妨害作用の考察本田昌實
マイクロ・メインフレーム・リンク用統合操作環境佐々木茂
画面作成データ・ストリーム変換プログラムの評価庭山宣幸
知識支援による設計ツール——ESP.....J. F. King, E. M. Hushebeck
ラビッド・プロトタイプングを支援する
知的ソフトウェア開発環境.....S. Pontecorvo, J. Krohnfeldt
BIBLIO——再使用可能コード・ライブラリ・システムK. L. Brusio

カナ漢字変換システム Micro JASTY の開発	吉田正行
意味的なつながりを考慮した接尾語辞書の作成	新谷隆之
DS 7 におけるコンカレント CP/M	阿部比呂志
TECHNOLOGY TREND	
Lisp マシンのウインドウ・システム	大田一久
MHS メッセージ通信システム	佐藤茂夫

第 14 号 (1987 年 8 月)★ 特集: MAPPER システム

論文 MAPPER 概論	川本光一
情報化社会のシステム開発と MAPPER の有効性	日置伸一
MAPPERKIT におけるデータ辞書	井上比佐乃
中規模製造業における MAPPER 5 の活用事例	A. E. Shelton
NASA における MAPPER 活用事例	
—MAPPER と DMS データベースを用いた	
資源アカウンティングおよび利用管理システム	
.....	F. Lovato, M. W. Mickelborough
トランザクション処理システムへの MAPPER の適用	小西徳雄
MAPPER によるユーザ・リアルタイム・システム事例	浦 伸一
MAPPER における排他制御とリカバリ方法の考察	岡島伸介
人工知能と MAPPER	
—Northwest 航空における事例	工藤隆司
MAPPER と光ディスク	
—花王(株)における事例	岩佐一夫

第 15 号 (1987 年 11 月)

論文 磁気ディスク制御装置における 14 バイト ECC の開発	上谷彌輔
イメージ・データの圧縮手法	河合昭男
衛星通信回線使用時の DCA リンク・レイヤ	
およびネットワーク・レイヤの伝送効率分析	宮坂順之
再使用可能コード・ライブラリ BIBLIO の開発	K. Brusio, P. Wright
リレーショナル・データベースの考察	
—整合性制約を中心として	原 潔
ネットワーク型データベース (DMS 1100) から	
リレーショナル・データベース (RDMS 1100) への変換	大内 忍
COSTAR (ATD) によるデータベース・リカバリの	
運用自動化の実現	小林俊平
各種バーコードのパーソナル・コンピュータによる印書実験	森 宗正
技術動向	
ICAI の技術動向	橘田 明, 右近 豊
データベース言語 SQL の標準化動向	原 潔

第16号 (1988年2月)*

論文 汎用有限要素法プリ・ポストプロセッサ APPEX中村暢樹, 野田安則
N 辺ボカシ面創成機能の開発藤井 省
汎用 CAD/CAM システム UNICAD®/SURFACE
における3次元編集設計新田光義
枠組壁工法住宅における構造の自動設計システム渡辺 寛
ソフトウェア設計法の味見峰尾欽二
JSD 仕様の直接実行加藤潤三
24 時間運転村田豊彦

技術動向
グローバル・トレーディング・システムの概要伊川 望

▶ 技報編集委員会

委員長 柳生孝昭

副委員長 米口 肇

委員 飯塚伊三雄, 岡井功雄, 久保田俊雄
佐野和義, 新野清嗣, 鈴木 勲,
内藤 聰, 永田利地, 西原良一,
野本雄一, 藤田康範, 前田英次郎,
村井啓一, 吉兼晴雄, 鷲尾 武, 朝倉
文敏, 駒崎洋介

▶ 編集制作担当

技術情報サービス部 青柳幸久, 丹野敬子

● Editorial Board

T. Yagiu (Chairman)

H. Yoneguchi (Vice Chairman)

I. Iizuka, I. Okai, T. Kubota K. Sano,

K. Shinno, I. Suzuki, S. Naitou,

T. Nagata, R. Nishihara, Y. Nomoto,

Y. Fujita, E. Maeda, K. Murai,

H. Yoshikane, T. Washio, F. Asakura,

Y. Komazaki

● Editorial Staff

Y. Aoyagi, K. Tanno

(Technical Information Services Dept.)

ISSN 0914-9996

技 報

UNISYS TECHNOLOGY REVIEW

Vol. 8 No. 4 (No. 20)

発行日 平成元年2月28日
編集人 柳生孝昭
発行人 富田和夫
発行所 日本ユニシス株式会社
東京都港区赤坂2-17-51 〒107
TEL(03)585-4111 (大代表)
頒布価格 1,500円
印刷所 三美印刷株式会社

禁無断複製転載

© Nihon Unisys, Ltd. 1989

UNISYS

ユーザーニーズに的確に応えます
トータル・システム・インテグレーター、ユニシス

90年代を透視して、鮮烈に浮上。

大規模トランザクション処理に対応します。画期的なXTPA[®]の採用で、さらにパワーアップ。いま、企業に必要とされるのは、90年代のスピードとパワーに対応できる潜在力をもった戦略情報システムの確立です。こうした時代の要求に、応えて登場したのがUNISYS 2200/600シリーズ。画期的なアーキテクチャXTPAの採用により、膨大なトランザクション処理にもゆうゆう対応。高いコストパフォーマンスと信頼性を大きく向上させました。*XTPA(EXTENDED TRANSACTION PROCESSING ARCHITECTURE・拡張トランザクション処理体系)従来機種との6倍以上の処理能力を提供。

先進技術を駆使。
超大型汎用コンピュータ



UNISYS 2200/600シリーズ

ユニシスから、新登場

日本ユニシス株式会社 本社 東京都港区赤坂 2-17-51 〒107 電話03-585-4111(大代表)