

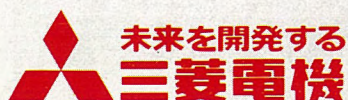
MITSUBISHI DENKI GIHO

三菱電機技報



Vol.58 No.6

コンピュータソフトウェア技術特集



6
1984

コンピュータソフトウェア技術特集

目次

特集論文

ソフトウェア品質管理標準要綱.....	1
桜井俊一・犬山博満・佐伯正夫	
産業応用向けに拡張したPROLOG.....	5
芥川哲雄・太細 孝・荻野敬迪・秋田興一郎・田村敏子	
漢字PROLOG処理系.....	9
太細 孝・向井国昭・鈴木克志・伊草ひとみ・佐藤裕幸	
プログラム自動検証システム.....	14
藤田 博・堀内謙二	
ソフトウェア開発のためのPWSシステム.....	18
内藤俊文	
マイクロプロセッサのソフトウェア開発支援システム.....	24
藤掛 遵・西田親生・福原直巳・長井秀憲	
基本ソフトウェア開発におけるC言語テストベッドの利用.....	28
大江信宏	
リアルタイムUNIX "OS60/UMX"	31
小川義高・臼井澄夫・高畑泰志・鈴木 裕・高橋良岳	

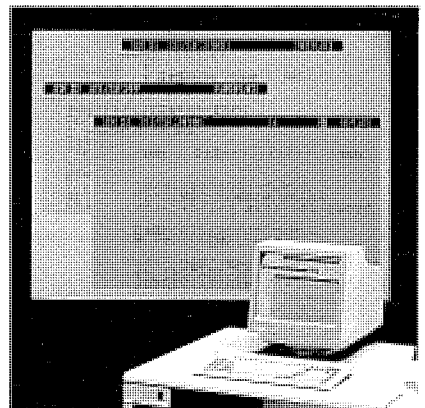
普通論文

燃料電池発電プラント制御システム.....	35
佐々木 明・水本洋一・日比野 学・大西俊一郎	
火力発電プラントの予防保全システム《MELRAP-T》.....	41
高田信治・広崎吉彦・明神俊治・仁科秀夫・富田義信	
姫路市水道局納め上水道配水コントロールシステム.....	45
駒田俊秀・阿藤隆英・半田 哲・福島勇二・中島弘善	
新規格準拠三相誘導電動機《スーパーライン J, Kシリーズ》.....	50
柳内芳彦・前原利昭・鈴木好弘・立見広光	
三菱オフィスターミナル M5000シリーズ.....	55
塚本久雄・守川修平・渡辺 透・田中雄三・佐立良夫	
3.5インチマイクロフレキシブルディスク装置.....	62
今村恒浩・佐々木 勝	
特許と新案.....	67, 68
ディスプレイ装置	
バス占有要求調停方式	
スポットライト	
大電力GTOサイリスタシリーズ.....	(表3)

表紙

《マルチスクリーンエディタ》

マルチスクリーンエディタはMELCOM 70 MX/3000のオペレーティングシステム OS 60/UMXのもとで動作するプログラム開発用のエディタです。マルチスクリーンで使えるほか、熟語変換、辞書機能付き日本語入力、カラー表示、アプリビエーション、プレイバック、コマンドスタック、たて／よこモード、インデンテーション、画面間のカット及びペーストなど豊富な機能をそなえています。OS60/UMXの、日本語が使えるFORTRAN 77やPROLOGでプログラミングするときに威力を発揮します。



アブストラクト

ソフトウェア品質管理標準要綱

桜井俊一・犬山博満・佐伯正夫

三菱電機技報 Vol.58・No.6・P1～4

計算機ソフトウェアの品質管理標準要綱を作成した。この要綱はソフトウェアの品質管理を有効かつ経済的に実施するための方針・手順を規定したものであり、方針管理、各開発段階とチェックポイント、品質管理組織、デザインレビュー、クレーム処理などからなっている。

本稿では、この標準要綱の内容概略及びこの標準要綱を適用することによって得られた成果を紹介する。

ソフトウェア開発のためのPWSシステム

内藤俊文

三菱電機技報 Vol.58・No.6・P18～23

この論文では、基本ソフトウェアの生産効率向上のための、設備環境とソフトウェア生産形態について《MELCOM-COSMO 900 II》を、ホスト電算機として構築したPWS（プログラミングワークステーション）システム及びソフトウェアツール データベースなどの開発支援サブシステムについて紹介する。

産業応用向けに拡張したPROLOG

芥川哲雄・太細 孝・荻野敬迪・秋田興一郎・田村敏子

三菱電機技報 Vol.58・No.6・P5～8

《MELCOM 350-60》を対象に産業応用向けに拡張したPROLOGを開発した。このPROLOGは、DEC-10 PROLOGの機能をほぼ包含している上に、独自の拡張機能として、グラフィックス基本機能、プロセス入出力機能、別タスクとの通信機能などを持っているため、従来の科学技術計算や計測・制御との仕事の分担が効率よく行える。産業応用向けの高度な知識情報処理システムを構築する際の有用な言語手段となるものである。

マイクロプロセッサのソフトウェア開発支援システム

藤掛 遵・西田親生・福原直巳・長井秀憲

三菱電機技報 Vol.58・No.6・P24～27

マイクロプロセッサの利用が急速に拡大しつつある。これに伴ってマイクロプロセッサのソフトウェア開発の生産性向上、信頼性向上が重要な課題となってきている。このため、システム移植性に富んだ“C”言語をベースとした8086マイクロプロセッサ用ソフトウェア開発支援システムを構築した。このシステムは、《MELCOM-COSMO 900 II》上でホスト処理するクロスシステムである。このシステムの特長、開発技術について紹介する。

漢字PROLOG処理系

太細 孝・向井国昭・鈴木克志・伊草ひとみ・佐藤裕幸

三菱電機技報 Vol.58・No.6・P9～13

知識情報処理に適した言語として注目を集めている述語論理形プログラミング言語PROLOGの処理系（漢字PROLOG）を、当社汎用計算機《MELCOM-COSMOシリーズ》上に開発した。述語論理記述の基本機能に加えて、漢字や、かな文字機能、妥当な処理速度、及びプログラミング環境が実現されており、日本語情報処理や機械翻訳をはじめとする知識情報処理分野において、基本ツールとして使用することができる。

基本ソフトウェア開発におけるC言語テストベッドの利用

大江信宏

三菱電機技報 Vol.58・No.6・P28～30

オペレーティングシステムなどの基本ソフトウェア及び周辺ソフトウェア、ユーティリティなどを開発するにあたり、その多くをシステム記述言語“C”で作成し、そのテストの段階でC言語テストベッドを利用することにより開発の効率化を図った。基本ソフトウェア開発におけるC言語テストベッドの利用方法と実際に利用した上での評価及び今後利用する上での注意点を紹介する。

プログラム自動検証システム

藤田 博・堀内謙二

三菱電機技報 Vol.58・No.6・P14～17

プログラムの虫取りや正しきの検証は、依然としてほとんど人の手にゆだねられている。検証には一般に多大な手間と時間を要する。そこで、この作業を計算機によってできるだけ自動化しようとする研究が進められている。検証の過程では、プログラムの満たすべき性質を定理として証明することが必要となる。

この論文は、Lispなどの関数形言語によって書かれたプログラムの満たすべき性質を検証する、定理証明システムについて述べる。

リアルタイムUNIX “OS60/UMX”

小川義高・白井澄夫・高畑泰志・鈴木 裕・高橋良岳

三菱電機技報 Vol.58・No.6・P31～34

高性能の実時間処理を行うと同時に、端末ユーザーに対しては操作性に優れたソフトウェア環境を提供するオペレーティングシステム OS60/UMXを開発した。これは、ソフトウェア開発用あるいはワークステーション用として高い評価を受けているUNIXオペレーティングシステムを、《MELCOM 350-60》計算機シリーズの実時間処理用オペレーティングシステムOS60に組み込んで一体化したものである。

Abstracts

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 18-23 (1984)

A Programming Workstation System for Software Development

by Toshifumi Naito

A number of improvements have been made in hardware environments and software-production schemata to increase the productivity of basic software development. The article introduces a programming workstation (PWS) system and development-support subsystem for the software tools, data bases, etc. produced using a MELCOM-COSMO 900II as the host computer.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 1-4 (1984)

A Standard Program for Software-Quality Control

by Shun'ichi Sakurai, Hiromichi Inuyama & Masao Saeki

The authors have developed a standard quality-control program for software. The program specifies policies and procedures for effective and economical control of software quality, and covers policy implementation, checkpoints at each phase of development, the organization of quality control, review of the software design, and any required corrective action.

The article summarizes this standard program's constituent elements and discusses the results achieved.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 24-27 (1984)

A Microprocessor-Software Development-Support System

by Jun Fujikake, Chikao Nishida, Naomi Fukuhara & Hidenori Nagai

The rapidly expanding use of microprocessors presents an urgent need for improved productivity and reliability in microprocessor software. The authors have developed a software development-support system for the 8086 microprocessor based on the highly flexible C-language. It is a cross-compiler system using a MELCOM-COSMO 900II as the host processor. The article introduces the system features and development technology.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 5-8 (1984)

Enhanced PROLOG for Industrial Applications

by Tetsuo Akutagawa, Takashi Dasai, Takamichi Ogino, Koichiro Akita & Toshiko Tamura

The authors have developed an enhanced version of PROLOG for industrial applications with the MELCOM 350-60 computer system. In addition to the original DEC-10 PROLOG functions, this version incorporates a series of unique expansion functions including basic graphics functions, process I/O functions, and task-to-task communications functions for better allocation of tasks, such as scientific calculation and measurement and control, than was previously possible. These advances make the new language revision a major asset in constructing high-level knowledge/information processing systems for industrial applications.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 28-30 (1984)

The Utilization of a C-Language Testbed for the Development of Basic Software

by Nobuhiro Oe

Basic and peripheral software utilities for operating systems are often developed using the C declarative language. The author has successfully employed a C-language testbed for software testing to increase software-development efficiency.

The article discusses methods, results, and special considerations in using the C-language testbed for developing basic software.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 9-13 (1984)

A Kanji PROLOG Programming System

by Takashi Dasai, Kuniaki Mukai, Katsushi Suzuki, Hitomi Igusa & Hiroyuki Sato

PROLOG, a predicate-logic programming language, has been drawing widespread attention for its effectiveness in knowledge/information processing. Using the general-purpose MELCOM-COSMO computer series, the authors have developed a PROLOG processor called *Kanji* PROLOG. Together with the basic predicate logic, *Kanji* PROLOG includes the functions required for *kanji* and *kana* characters, suitable processing speeds, and a full programming environment. The system is designed as a basic tool for use in Japanese information processing, machine translation, and other knowledge/information fields.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 31-34 (1984)

OS60/UNIX, a Real-Time UNIX System

by Yoshitake Ogawa, Sumio Usui, Yasushi Takahata, Yutaka Suzuki & Yoshitake Takahashi

The newly developed OS60/UNIX operating system provides high-performance real-time processing capabilities, gives excellent operability to the terminal user, and provides a superior software environment. The OS60/UNIX system is an integration of the UNIX system, with its established reputation for software development and workstations, into the OS60 real-time operating system of the MELCOM 350-60 computer series.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 14-17 (1984)

An Automatic Program-Verification System

by Hiroshi Fujita & Kenji Horuchi

Almost all program debugging and verification is still done manually. Program verification involves the process of checking that the program meets all its theoretical performance specifications. This generally requires a massive expenditure of time and effort, but research aimed at the fullest-possible automation of program verification is making progress.

The article describes one such verification system designed to confirm the performance of programs written in LISP and other functional programming languages.

アブストラクト

燃料電池発電プラント制御システム

佐々木 明・水本洋一・日比野 学・大西俊一郎

三菱電機技報 Vol.58・No.6・P35～40

燃料電池発電システムは、負荷追従性に優れ、起動時間も短く、高効率の新形電源として実現が期待されている。当社ではMW級りん酸形燃料電池発電システム開発の一環として、50kW-MKIIプラントを試作し自動運転試験を実施した。

今回は50kW-MKIIプラントを例として、制御対象と考えた場合のプラント特性、燃料電池の特長を最大限発揮するために必要な制御システムの構成・機能及び運転結果について報告する。

新規格準拠三相誘導電動機《スーパーライン J, Kシリーズ》

柳内芳彦・前原利昭・鈴木好弘・立見広光

三菱電機技報 Vol.58・No.6・P50～54

モータルの規格であるJIS及びJEM規格が、国際規格であるIECとの整合化を目的として、今回全面的に改訂された。当社では、これら新規格に準拠した新系列一般用三相誘導電動機を完成したので、その特長である信頼性の向上、小形軽量化、作業性の向上、保守の容易化、特性の向上などを中心にして、新系列モータルの概要を紹介する。

火力発電プラントの予防保全システム《MELRAP-T》

高田信治・広崎吉彦・明神俊治・仁科秀夫・富田義信

三菱電機技報 Vol.58・No.6・P41～44

昭和58年度に電算機を応用した火力発電プラント予防保全システム《MELRAP-T》を完成した。このシステムは、各発電ユニットの主要構成電機品ごとに固有の予防保全項目を定めて、予防保全実施履歴をベースとして、必要な予防保全項目と必要な実施時期をアウトプットするものである。電算機を利用することにより、多数のユニット・機器に横断的な予防保全情報が得られるようになり、信頼性と稼働率の向上に寄与できるようになった。

三菱オフィスターミナル M5000シリーズ

塚本久雄・守川修平・渡辺 透・田中雄三・佐立良夫

三菱電機技報 Vol.58・No.6・P55～61

ホストコンピュータのソフトウェア及びデータの資産を活用するためのオンライン端末としての機能、パーソナルコンピュータとしての機能、ワードプロセッサとしての機能及びオフィスコンピュータ相当のデータ処理機能を併せもった三菱オフィスターミナル M5000シリーズについて、その概要を述べる。

姫路市水道局納め上水道配水コントロールシステム

駒田俊秀・阿藤隆英・半田 哲・福島勇二・中島弘善

三菱電機技報 Vol.58・No.6・P45～49

上水道の分野において、広域にわたる配水施設の維持管理として需要家配水圧の安定化や需要予測に基づいた施設の計画的運用による省資源、省エネルギーが要請されている。

本稿では計算機を導入し、需要予測による配水計画運用ガイダンス、管網シミュレーションを用いた水圧制御ガイダンスなどによりこれらの要請にこたえた、姫路市水道局配水コントロールシステムについて紹介する。

3.5インチマイクロフレキシブルディスク装置

今村恒浩・佐々木 勝

三菱電機技報 Vol.58・No.6・P62～66

パソコン、OA機器の外部記憶装置として、8インチ、5.25インチのフレキシブルディスクドライブが飛躍的に伸びている。最近磁気ディスクをハードケースに収納し、ヘッドウィンドーにシャッタを設け、小形ながら記憶容量が従来と同等で、より使いやすくした3～4インチフレキシブルディスクが現れた。当社は大容量が得やすい3.5インチフレキシブルディスクが使えるマイクロフレキシブルディスクドライブを開発したので、その特長などを述べる。

Abstracts

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 50-54 (1984)

Super Line J and K: New Series of General-Purpose Induction Motors

by Yoshihiko Yanai, Toshiaki Maehara, Yoshihiro Suzuki & Hiromitsu Tatsumi

JIS and JEM motor standards have recently been thoroughly revised in line with the international standards established by the IEC. Mitsubishi Electric has developed two new lines of general-purpose three-phase induction motors conforming to the new standards. The article introduces these new motor lines and describes design advances permitting greater reliability, reduced size and weight, improved operability, easier service, and better electrical characteristics.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 35-40 (1984)

The Development of a Control System for a Fuel-Cell Power Plant

by Akira Sasaki, Yoichi Mizumoto, Manabu Hibino & Shun'ichiro Onishi

Excellent load-following characteristics, short startup time, and high efficiency make fuel-cell power plants a promising new power source. As an intermediate step toward development of a megawatt-class phosphoric-acid fuel-cell power system, Mitsubishi Electric has produced the Mark-II 50kW prototype plant and tested its performance characteristics under automatic control.

The article describes the operation characteristics of the plant, discusses the control-system configuration and operation parameters required to optimize plant performance, and reports on the results of the test operation.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 55-61 (1984)

The M5000 Series of Office Terminals

by Hisao Tsukamoto, Shuhei Morikawa, Toru Watanabe, Yuzo Tanaka & Yoshio Sadachi

In addition to functioning as on-line terminals to optimize host-computer software and data-resource utilization, the M5000 series of office terminals can operate independently as personal computers and word processors with the data-processing capability of small-business computers. The article summarizes the features of this new series of office terminals.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 41-44 (1984)

MELRAP-T, a Program for Upgrading the Reliability and Availability of Thermal-Power Plants

by Nobuharu Takata, Yoshihiko Hirotsaki, Shunji Myojin, Hideo Nishina & Yoshinobu Tomita

In fiscal 1983, the development of the MELRAP-T computerized preventive-maintenance system for thermal-power plants was completed. In this system, individual preventive-maintenance (PM) items for each of the principal electrical components is specified by computer output in accordance with the test items and time intervals previously established. The use of a computer system permits collation of PM data on multiple units or devices, contributing to improved reliability and availability.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 62-66 (1984)

A 3.5-Inch Micro Flexible-Disk Drive

by Tsunehiro Imamura & Masaru Sasaki

Personal computers and office-automation equipment now make extensive use of 8-inch and 5.25-inch flexible-disk drives as external storage media. Recently, equivalent storage capacities have been achieved on smaller, easier-to-use 3- to 4-inch disks by packing them in a hard case with a shutter covering the head window. Mitsubishi Electric has developed a 3.5-inch micro flexible-disk drive designed to fully exploit the improved storage density of the new disks. The article describes this new drive and its features.

Mitsubishi Denki Giho: Vol. 58, No. 6, pp. 45-49 (1984)

A Water-Supply Network-Control System for the Himeji Waterworks Bureau

by Toshihide Komada, Takahide Ato, Satoshi Handa, Yuji Fukushima & Hiroyoshi Nakajima

To provide a reliable water supply requires control and maintenance of a large amount of distribution equipment spread over a wide area. A stabilized water supply-pressure and the use of water-demand prediction is essential for efficient use of this equipment to save energy and water resources.

The system developed for the Himeji Waterworks Bureau uses computers to control the water-supply network. Demand predictions are used to generate water-dispatching planning guidance and distribution-network control simulations which, in turn, are used to produce supply-water pressure-control guidance to match demand. The article reports on the system and its features.

1. ま え が き

ソフトウェア品質管理の基本的な考え方は、一般工業製品で確立している次のような品質管理手法を導入して、ソフトウェアの特性に合わせて運用することである。

- ・標準類を整備して実施する。
- ・品質管理 (PDCA) サイクルを回す。
- ・品質データを収集し統計的手法により解析する。

従来からこの考え方を基盤としてソフトウェア品質管理（以下、品質管理）を主に次のような具体的方策を展開することによって改善・強化してきた。

(1) 設計段階を中心に、品質の源流管理を徹底する。

(a) ソフトウェアの標準類を設計段階を中心に充実・整備し、その実施を徹底する。

(b) 開発の各段階ごとに品質審査や第三者検査を実施する。

(2) 品質管理部門をソフトウェアの品質管理サイクルの中核に位置づけ、品質管理サイクルをよく回す⁽¹⁾。

(3) 開発の各段階で収集する品質データを蓄積・解析するシステムを充実・整備し、収集した品質データを活用する⁽²⁾。

このように品質管理を改善・強化してきた結果、ソフトウェア製品の品質を大幅に改善することができた。この実績をフィールドにおけるソフトウェア製品の障害件数で示すと、昭和53年度を100とすると昭和56年度には45になった。

一方、ソフトウェアのトラブルによる業務障害（バンキングシステムや電話交換システムなどでの障害）が大きな社会問題になっている。更に、パーソナルコンピュータの普及により、ソフトウェアが多種多様な場所で不特定多数の人達によって使用され、ソフトウェアの品質の悪さが引き起こす問題を無視できない社会状況になっている。このような社会状況から、高品質ソフトウェア製品を開発するために品質管理を更に改善・強化する必要に迫られている。このため新たにソフトウェア品質管理標準要綱を作成し、品質管理の高度化を図っている。

なお、この論文で使用する用語を次のように定義する。

- (1) 障 害 : ソフトウェア製品に潜在する誤りが表面化して規定の機能を失うこと。
- (2) クレーム : フィールドにて発生するユーザーからの苦情。

2. 標準 SQCP による品質管理の高度化

品質管理を更に高度化するために、現行の品質管理を集大成して普遍化した。すなわち、方針管理を主柱として次のような施策を有機的に結合して品質管理標準となるソフトウェア品質管理標準要綱（以下、標準 SQCP : Software Quality Control Program）を作成し、その遵守を徹底した。

- (1) 適切な品質管理段階を設定する。
- (2) 品質管理サイクルをよく回す体制を充実する。
- (3) デザインレビューを充実する。

表 1. 当社の標準 SQCP と IEEE の SQAP

標準 SQCP	IEEE の SQAP
—	参考文献
方針管理	—
各開発段階とチェックポイント	—
品質管理組織	管 理
デザインレビュー	審査と監査
クレーム処理	問題報告と解決処理
構成管理	構成管理
コード管理	コード管理
媒体管理	媒体管理
外注管理	購入品、外注品管理
文 書 化	文 書 化
—	標準、規定
生産技術・ツール	ツール、技術、方法論
—	記録の収集、維持、保存

(4) 品質データ（クレームに関するデータなど）に基づき再発防止対策を徹底する。

この標準 SQCP の項目は、方針管理、デザインレビュー、組織などとなっている。なお、ソフトウェアの先進国であるアメリカでも IEEE が SQAP (Software Quality Assurance Plans)⁽³⁾ を作っている。ここで紹介する標準 SQCP とこの SQAP とは考え方に類似した点が多い。両者の比較を表 1. に示す。

当社計算機製作所では、現在、標準 SQCP を規範としてソフトウェアの各開発プロジェクト（以下、プロジェクト）の特性に合わせた個別のソフトウェア品質管理要綱（以下、SQCP）を設定して品質管理を実施している。この SQCP はプロジェクトの開発着手に先立って品質管理部門が立案し、開発計画作成時に開発部門と協議して設定している。

標準 SQCP を作成し活用した結果、次に示す効果が出てきている。

- (1) 源流管理の意味が、より明確になり源流管理が徹底できた。
- (2) 品質管理活動が標準化され円滑になった。
- (3) 統一的な整理しやすい品質データを得た。
- (4) 品質方針が関係者全員に徹底できた。

3. 標準 SQCP の内容

標準 SQCP では、品質管理を実施するために必要な項目だけでなく、その考え方や管理の進め方についても明確にして使いやすいつものにしている。以下に、標準 SQCP の主要項目についてその概要を紹介する。

* 計算機製作所

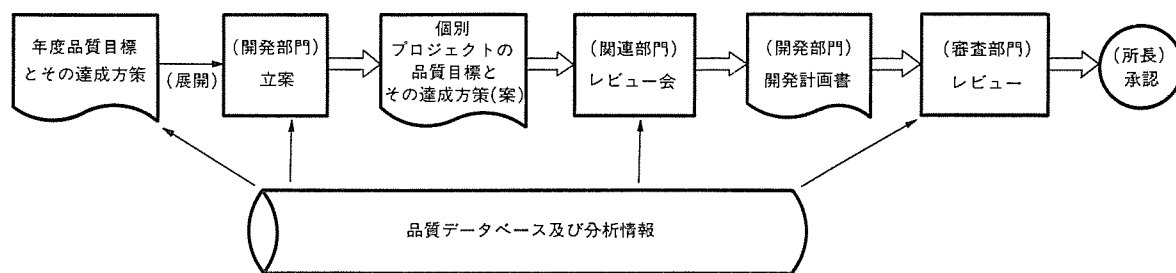


図 1. 個別プロジェクト品質方針の設定過程

3. 1 方針管理

以下に示す品質方針の設定及び方針展開を通じて方針管理を推進する。

3. 1. 1 品質方針の設定

長期計画に基づいて、部レベルから課レベルへと品質方針（品質目標とその達成方策）をより具体的なレベルに展開する。また、長期品質向上方針、年度品質方針及びプロジェクト単位の品質方針を段階的に設定する。設定にあたっては、関連部門がよく協議し合意に達することが重要である。

ここで、一例としてプロジェクト単位の品質方針設定のやりかたについてその概要を紹介する。図 1. にその概念を示す。

(1) まず、年度品質方針に沿って対象ソフトウェア製品の品質目標（品質要素ごとの目標）と、その達成方策を開発計画段階で立案させる。この妥当性を、レビュー会（レビュー委員は開発部門、SE 部門及び審査部門から召集する）で検討した上で、開発計画書にまとめさせる。次いで、この開発計画書は審査部門（営業部門、経理部門及び品質管理部門）のレビューを経て承認を受ける。

(2) 上記レビューでは、対象製品の母体製品や同種製品の品質データを分析して得た情報を活用して、検討及び改善指摘を行う。その例を次に示す。

(例 1) クレームの内容・件数などの分析結果を活用して要求満足度をレビューする。

(例 2) 試験で誤りを検出できなかった理由などのデータ分析結果を活用して、試験計画をレビューする。

(例 3) 誤りが入り込んだ作業段階を示すデータ、レビュー実施データ（レビューの方法、所要工数、効果などに関するデータ）などの分析結果を活用して、レビュー計画をレビューする。

3. 1. 2 方針展開

各開発段階で、段階的に展開された品質目標とその達成方策を品質審査などを通じてレビューする。このとき、品質目標達成度を予測し必要があれば目標達成方策を見直す。見直した目標達成方策の実施状況は、定期レビューなどでフォローし実施徹底させる。このようにして、所期の品質目標を達成できるよう方針展開を推進する。

3. 1. 3 品質目標

品質目標と、その達成度評価結果のフィードバックのやりかたについて紹介する。

(1) 品質目標は次に示す品質要素ごとに設定する。

- (a) 要求仕様の満足度（機能満足度、性能満足度など）
- (b) 信頼性
- (c) 保守性

(2) 品質目標は段階的に次の 3 種類を設定する。

- (a) 長期品質向上目標

長期計画に対応して設定するもので、毎年見直しを行う。

(b) 年度品質目標

長期品質向上計画の中間年度としての目標を設定する。目標達成度の評価は毎月実施し、品質委員会などの場を通じて評価結果を開発部門にフィードバックする。また年度末での評価結果は、品質評価報告書にまとめて開発部門にフィードバックする。

(c) プロジェクト単位の品質目標

年度品質目標を達成できるよう、個々のプロジェクトに対してもその特性に応じて品質目標を設定する。目標達成度の評価は、製品出荷後一定の監視期間を終了した時点で実施する。この評価結果も品質評価報告書にまとめて開発部門にフィードバックする。

(3) 上記の品質評価報告（目標達成度評価など）を次のように活用し、品質向上に結びつける。

(a) 開発部門での活用

重点課題の明確化、組織・体制・制度の改善、個別プロジェクトでの改善など。

(b) 品質管理部門での活用

品質管理活動の改善、品質管理方式の改善など。

3. 2 各開発段階とチェックポイント

開発段階を作業単位に細分化し、工程を可視化する。この可視化した工程を品質と対応させて、いくつかの品質管理段階に分割し、品質審査をするのに適切なチェックポイントを得られるように品質管理段階を設定した。これを表 2. に示す。

この品質管理段階ごとに、品質保証を目的として品質審査を実施

表 2. 品質管理段階

開 発 段 階	品 質 段 階	品 質 管 理 段 階
調 査 ・ 企 画	品 質 設 定	調 査 ・ 企 画
計 画		計 画
外 部 設 計	品 質 決 定	設 計
内 部 設 計		
製 造		製 造 ・ 試 験
試 験		
検 査	品 質 保 証	検 査
仮 登 録		登 録
登 録		
出 荷	品 質 維 持	出 荷
保 守		保 守

表 3. 計画段階での品質審査

品質管理段階	審 査 目 的	審 査 時 期	審 査 部 門	承 認 者	審 査 内 容	審 査 対 象 物	審 査 方 法
計 画	基本仕様、納期、費用、品質目標などのオーソライズ	開発計画原案完了時又は開発計画書提出時	営 業 部 門 経 理 部 門 品質管理部門	所 長	基本仕様、納期、費用、品質目標などの妥当性	開 発 計 画 書	開発同審査会議 レビュー会

する。どのような品質審査を実施するかについては、5W1Hが明確になるよう規定している。例えば、計画段階での品質審査については表3.のようである。

3.3 品質管理組織

品質管理段階を踏まえ、品質管理のPDCAサイクルを図2.のように定めた。品質管理部門は、このPDCAサイクルの中枢に位置し、品質目標の達成度評価結果や改善施策などを品質委員会などあらゆる場を通じて開発部門にフィードバックする。また、重大なクレームに対しては、関連部門を召集して対策会議を行い応急対策を推進する。このほか、ソフトウェア小集団活動を指導・推進するなど、種々の活動を有機的に結びつけてPDCAサイクルがよく回るように活動する。この品質管理活動組織の概要を図3.に示す。

3.4 デザインレビュー

デザインレビューは、各開発段階において次のことを目的として実施する。

- (1) 誤りの検出
- (2) 品質目標の達成度予測
- (3) 仕様の確認
- (4) 教 育

レビュー対象は、各開発段階で出力されるすべての中間生産物（主に仕様書類）である。レビュー方式には、大別して開発部門内レビューと第三者レビュー（品質管理部門による品質審査など）とがある。レビュー実施にあたっては、各仕様書ごとに用意した標準チェックリストを使用しレビュー効果を高めるようにしている。

デザインレビューを通じて品質目標達成度を予測した結果、所期の品質目標を達成するために達成方策を見直す必要がある場合は、品質管理部門は対策を指示するとともに改善策を提示する。更に、この改善対策と処置状況を月例工程会議でレビューし、品質目標を達成するよう改善を徹底させる。このようにして、早期に目標達成のための手を打っていく。

なお、上記の活動においては各種品質データ（レビュー実施データ、仕様書評価データ、試験に関するデータなど）を標準様式を用いて収集し活用する。これを、対象製品の品質評価に使用するだけでなく、母体製品や同種製品の品質データとも合わせて、レビューや試験の重点項目を洗い出すために使用するなど、品質目標達成方策の見直しにも活用する。また、これらの品質データは、開発標準類や標準S

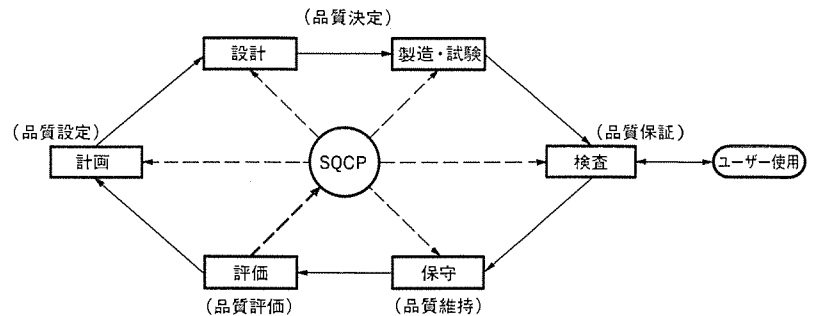


図 2. ソフトウェア品質管理サイクル

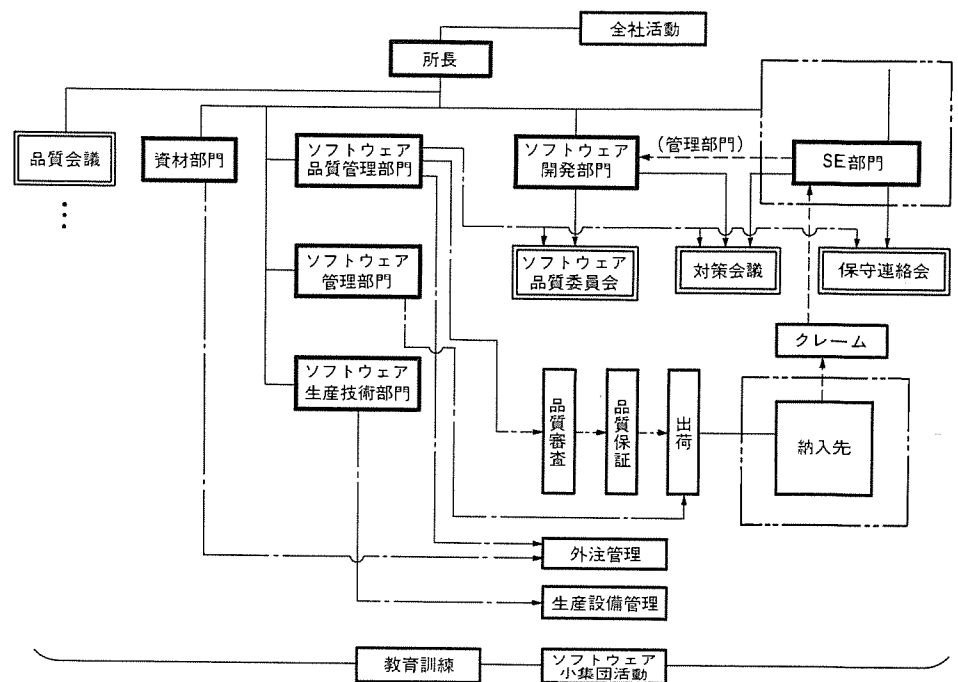


図 3. 品質管理活動組織図

QCPを改善していくためにも、データベース化して蓄積しておく。

3.5 クレーム処理

ユーザーからのクレームを、「漏れなく」、「早く」、「正確に」処理するために、クレーム処理の督促、クレーム処理日数の目標管理及び修正ミス率の目標管理活動を推進する。また、応急対策の審査や再発防止対策の推進などを行う。以下に応急対策と再発防止対策についてその概要を紹介する。

3.5.1 応急対策

保守部門が検討した応急処置・対策を、品質管理部門はレビューしその妥当性をチェックする。重大クレームに対しては、関連部門を召集して対策会議を行い応急対策をとりまとめる。

3.5.2 再発防止対策

具体的な再発防止対策は、ソフトウェア小集団活動などの場で開発部門

に検討させ、その成果を標準類の改善に結びつける。品質管理部門は、小集団活動の指導・推進役を果たすとともに、クレーム処理を通じて収集する各種品質データをベースにして品質目標達成度評価や要因分析を行い、その結果をフィードバックすることによって再発防止対策を推進する。以下にその概要を紹介する。

(1) 品質目標達成度の評価方法について

品質データベースに蓄積しているクレームデータと障害データとをベースに、次のような方法で品質目標に対する実績を評価する。

- (a) 信頼性については、障害件数（生産ステップ数でノーマライズした障害件数）などをもとに評価する。
- (b) 保守性については、クレーム処理日数、修正ミス率（修正ミス件数を修正件数でノーマライズしたもの）などをもとに評価する。
- (c) 要求満足度については、クレーム件数、クレーム照会区分などをもとに評価する。

また、各種品質データ及び生産管理データを活用して多角的に詳細分析（品質要因分析など）も実施する。

なお、これらの作業を支援するシステムとして、ソフトウェア品質評価支援システム“SQUARE”(Software Quality and Reliability Evaluator)があり、これを活用して迅速に評価・分析作業を行う。

(2) 各製品の品質向上のためのフィードバック

品質目標に対する実績を毎月監視し、限界値への接近／超過などの問題があれば必要に応じて詳細分析を実施して、対策検討を指示するとともに改善策を提示する（追試験や改版の指示、試験やレビューにおける重点項目の提示など）。このようにして、所期の品質目標を満足するよう対象製品の品質向上を推進する。

(3) 開発標準類や標準 SQCP の改善のためのフィードバック

一定の監視期間（標準として登録後6か月）を経過した製品について、品質目標に対する実績を評価するとともに詳細分析を行う。また、それらの評価、分析を積み上げて、設計・試験方法、レビュー方法、チェックリスト、審査方法、品質評価基準などを分析して問題点などを洗い出す。これらの結果を開発部門や品質委員会に報告し、開

発部門での品質向上活動を推進するとともに品質管理部門自身でも改善検討して、各種標準類の改善を推進する。

4. む す び

以上、標準 SQCP と、標準 SQCP を活用した品質管理、について紹介した。

標準 SQCP を作成したことにより、源流管理を徹底するとともに、品質方針を関係者全員に認識させ、方針展開を徹底するというねらいが達成でき、ソフトウェア製品の高品質化に役立ちつつある。その成果の一端を次に示す（昭和53年度を100とした指数で示す）。

- ・ 障害件数が減少した。

53年度=100	}	品質管理の改善・強化
54年度= 82		
55年度= 63		
56年度= 45		
57年度= 20	標準 SQCP の適用	

今後、標準 SQCP を更に改善・強化し、より高品質なソフトウェア製品を開発するために、今後とも次のことを主要課題としてとらえて努力していきたいと考えている。

- (1) 標準 SQCP の個々の内容（外注管理、第三者検査、生産技術・ツール）を更に充実する。
- (2) マイクロプロセッサのソフトウェアへの適用を推進する。

参 考 文 献

- (1) 桜井：ソフトウェア・フィールド故障データバンクの活用，日科技連第12回信頼性・保全性シンポジウム発表報文集，p. 471
- (2) 中山・佐伯：ソフトウェア品質評価支援システム“SQUARE”，三菱電機技報，57，No. 4，p. 321（昭58）
- (3) IEEE Standard for Software Quality Assurance Plans, ANSI/IEEE Std 730-1981

産業応用向けに拡張したPROLOG

芥川 哲雄*・太細 孝**・荻野 敬迪***・秋田興一郎+・田村 敏子*

1. ま え が き

近年 CAE や FA の分野でも EXPERT SYSTEM の試作が盛んになっており、LISP や PROLOG などの人工知能用言語が便利に使用され始めている。以下に紹介する《MELCOM 350-60》PROLOG は、特に産業応用を意識して、オンラインリアルタイム 向きに独自の機能拡張を行っており、この分野での知識工学システム構築に際して有力な言語手段となすべく設計された。その機能的特長を要約すると図 1. に示すように、従来の科学技術計算や計測・制御との仕事の分担が効率よく行え、例えばプラントの監視・診断・ガイドといった高度情報処理システムへの応用が期待できる。

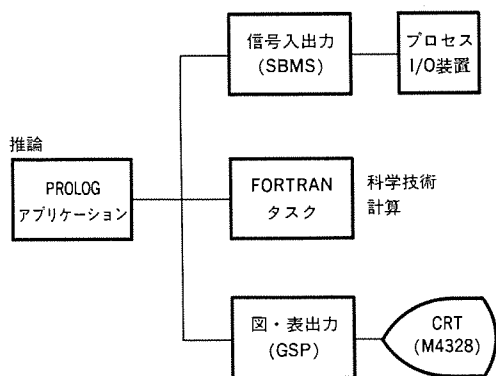


図 1. 機能概要

2. 言語仕様

《MELCOM 350-60》PROLOG の言語仕様は基本的に、エジンバラ版 DEC-10 PROLOG の仕様に準拠しているが、カタカナ、浮動小数点数、文字列などの機能が追加されている。シンタックスの表記も変数やリストなどを別にすれば、ほぼ DEC-10 PROLOG と共通になっている。ただし、用途が少ないか、又は処理負担が過大な機能は実現されていない。以下にこの PROLOG の言語仕様について説明する。

2.1 項の形式

PROLOG 言語における表記の基本単位は「項」と呼ばれ、次のように分類される。右側に表記例を付記する。

項	定数	整数	: 0, -99, 2'1011, X'F 07 F
		浮動小数点数	: 5.3 E-3, 10.5
		アトム	: ATOM, アトム, 'アトム'
		文字列	: "STRING", " " " "
	変数		: *X, *ヘンスウ, *
	複合項		: A(B, *C, 5)
	リスト		: <1, 2, 3>, < >

- ・整数 : $-2^{31} \sim 2^{31}-1$ の範囲の値を扱う。表記法には標準の 10 進のほか、2, 8, 16 進がある。
- ・浮動小数点数 : 絶対値が $5.398 \times 10^{-79} \sim 7.237 \times 10^{75}$ の範囲の

値を扱う。指数部のない表記も許される。

- ・アトム : 任意長のシンボル列からなる。
- ・文字列 : シンボル列を引用符「"」で囲んだもの。この処理系では文字列形を新設して効率的に扱っているので、DEC-10 PROLOG との互換性はない。
- ・変数 : アトムの直前に「*」を付加したもので、統合化により値を持つ。「*」だけからなる変数は、匿名変数と呼ばれ、変数であることが分かりさえすればよい箇所で使われる。
- ・複合項 : 構造を持った項で、ファンクタ及び引数の並びからなる。ファンクタにはアトムになるが、引数には再び複合項がなり得る。
- ・リスト : 複合項の特別な形式であり、複雑なデータ構造を表現できる。

2.2 プログラムの形式

PROLOG のプログラムは、ホーン節と呼ばれる論理式を並べたものである。ホーン節の理論上の基本形式は次の 3 種である。

ホーン節	事実	: P :-
	規則	: P :- Q ₁ , Q ₂ , ..., Q _m
	質問	: :- Q ₁ , Q ₂ , ..., Q _n

※但し、P, Q_i は前述の複合項である。また「:-」の左側を頭部、右側を本体、そして本体の各要素をゴールと呼ぶ。

- ・事実 : 頭部だけからなるホーン節で、「P は正しい」という意味を持つ。例えば、トリ (スズメ)、イロ (リンゴ、アカ) など。
- ・規則 : 「Q₁~Q_m が正しければ P は正しい」という意味。例えば、トブ (*X) :- トリ (*X) は、「トリはトブ」を表す。
- ・質問 : 本体だけからなるホーン節で、「Q₁~Q_n は正しいか」という意味を表す。事実と規則によって構成される PROLOG プログラムに対し、質問は実行を促す働きをする。

2.3 組み込み述語

従来形言語における組み込み関数と同じく、PROLOG には数多くの組み込み機能が用意されている。ユーザーは、これらを自由にプログラム中で使うことができる。《MELCOM 350-60》PROLOG では約 100 種類の組み込み機能が実現されている。

- ・入出力 (約 40 種) : 外部ファイル、周辺端末に対する入出力、デバッグ出力のログ機能など。
- ・算術 (約 10 種) : 四則演算、比較、代入、形変換など。
- ・文字列 (1 種) : 代入、但し関数機能として多種有り。
- ・便宜 (6 種) : プログラムの記述性、読解性向上のための機能。プログラムの AND 形実行指定、OR 形実行指定など。
- ・特別制御 (5 種) : プログラム実行制御のための CUT, REPEAT 機能及び IF-THEN-ELSE 形実行制御など。
- ・超論理 (14 種) : 形判定が約 3 分の 1、それにリスト一項間変換、高階述語呼出し (従来形言語における間接呼出し様の機能) など。
- ・環境 (約 20 種) : テスト/デバッグ、状態表示、モード切替など。

3. 処理方式

《MELCOM 350-60》PROLOG の処理系はインタプリタであり、移植性を高めるために基本的にC言語で記述され、一部組み込み述語がPROLOG自身で記述されている。またメモリ管理方式には、処理が簡潔である上、実行効率や拡張性の点で有利なコピー方式を採用している。TRO (Tail Recursion Optimization) は行っていない。以下に、この処理系のモジュール構成、及びコピー方式におけるメモリ管理について説明する。

3.1 モジュール構成

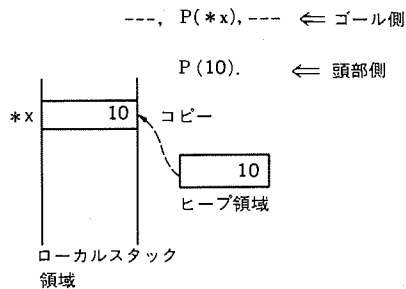
図2. がこの処理系のモジュールの構成である。Pモジュールはインタプリタ本体で、ホーン節の実行(統合化及びバックトラック)を制御する、COMMONモジュールには、大域定数、マクロ定義、組み込み述語の入口記述などが保持され、Pモジュールはここを参照しながら組み込み述語を呼び出したりする。

一方、Mモジュールはヒープ、及びスタック領域を使ってプログラム、及び動的情報を管理する。また今回の実現の大きな特長は、ユーザー定義の組み込み述語を用意できることにある。つまりユーザー各々が自分の望む組み込み述語の仕様を作ることができる。なおこの処理系のソースステップ数は約6,000行(C言語記述)、ロードモジュールサイズは約170Kバイト(但し、COMMON領域を除く)である。

3.2 コピー方式におけるメモリ管理

これまで開発された処理系は、DEC-10 PROLOG を初めとしてほとんどが構造共有方式を採用している。構造共有方式とコピー方式

非変数が1ワード



非変数が1ワード以外

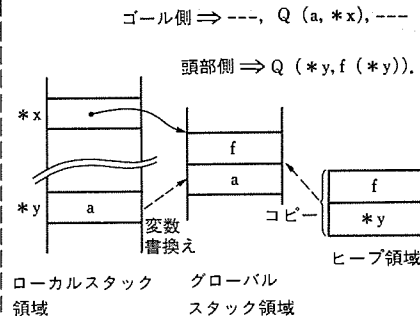


図4. コピーの方法

の主な違いは、構造を持ったデータを表すとき、前者は $\langle s, e \rangle$ (s : スケルトン, e : スケルトンに出現する変数の環境) という対(分子と云う)で表現する。一方、コピー方式ではスケルトンをスタック領域にコピーし、変数を対応する定数で書換える。このようにコピー方式では、スケルトンを共有しないためメモリ消費量は増加するが、スタック領域からヒープ領域へのチェーンは不要となり、したがって分子も不要となるので、メモリ管理は容易になる。両者の違いを要約すると図3. のようになる。

コピーが起るのは、ゴール側の変数と頭部側の非変数が統合化されたときであるが、今回の実現では図4. に示すように、非変数の大きさが1ワードであれば直接変数にコピーし、さもなくば非変数をグローバルスタックにコピーして、ゴール側の変数からそこへチェーンを張る方式とし、更にメモリ消費量の圧縮を図っている。

4. 拡張機能

《MELCOM 350-60》PROLOG の産業応用向けに拡張した独自の特長となっている三つの機能について、構造と使い方を説明する。このうちグラフィックスとプロセス入出力の二つは、各々 GSP (Graphic Subroutine Package) と SBMS (Sensor Base Management System) という既存の基本ソフトウェアを組み込み述語 FCALL により呼び出すことで実現する。3番目の別タスクとの通信機能には組み込み述語 FORK を用意している。

4.1 グラフィックス

マンマシンインタフェースとしてのグラフィックス機能は、汎用か産業用かの分野を問わず、ますます重要になってきている。GSP は ACM 「CORE」システムに準拠して作られているが、M4328 ディスプレイ装置を用いる場合、若干の機能の追加/削除がなされている。PROLOG プログラムでの述語の一般形は次のようになる。

FCALL FSUB (パラメータの並び)

ここで FSUB が GSP 中のサブルーチン名であり、パラメータの並びはそのサブルーチンの引数に対応する。例えば、現在位置を始点として指定位置 (100, 200) まで線分を描く GDRAW を使いたいときには、

....., FCALL GDRAW (100, 200),

とすればよい。図5. に簡単な図形を描くプログラムの例をあげるが、GSP を使う際の注意として、その入力引数は FCALL というゴールの実行時に具体的に決っていなければならない。

4.2 プロセス入出力

SBMS には表1. に示す7種類のサブルーチンがあるが、これらを P

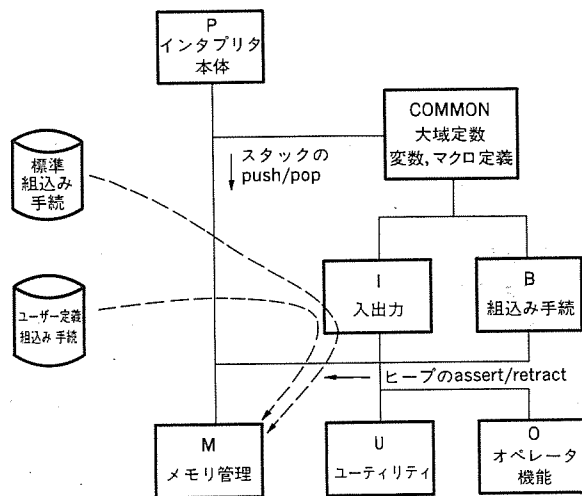


図2. モジュール構成図

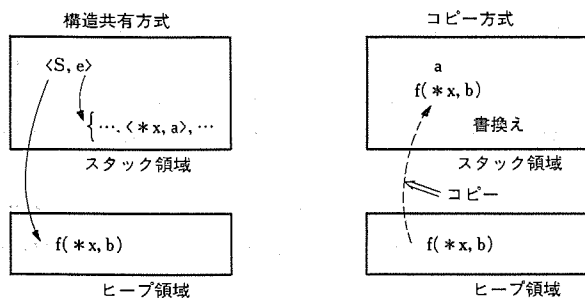


図3. 構造共有方式とコピー方式


```

INITIALIZE:—
  FCALL GINIT (41, 3, 2, X'F, IOBUF, 0, 0),
  FCALL GWINDOW (0. 0, 100. 0, 0. 0, 100. 0),
  FCALL GVIEWP (0. 0, 1. 0, 0. 0, 0. 74).
TERMINATE:— FCALL 'GTERM.
DRAW (*VALUE):— *VALUE:>800. 0.
DRAW (*VALUE):—
  *ADD IS *VALUE + 10. 0,
  FCALL GMOVE (*VALUE, 100. 0),
  FCALL GDRAW (0. 0, 0. 0),
  FCALL GMOVE (*VALUE, 0. 0),
  FCALL GDRAW (0. 0, 100. 0),
  DRAW (*ADD),
  FCALL GMOVE (*VALUE, 100. 0),
  FCALL GDRAW (*VALUE, 0. 0).
:—TTYWRITE ("CONSULT END").

```

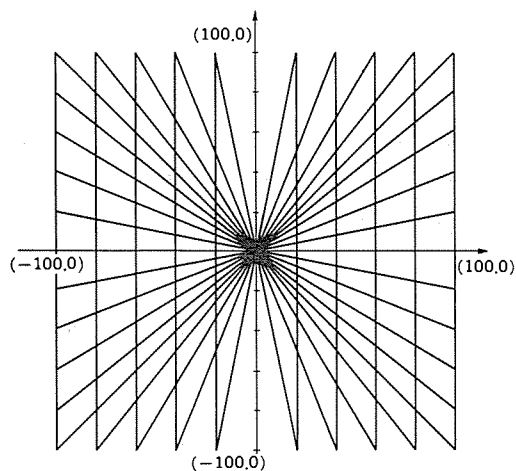


図 5. プログラム例

表 1. SBMS のサブルーチン

PSIN	汎用プロセス入力
PSOUT	汎用プロセス出力
PSSRED	直結専用プロセス入力
PSSWRT	直結専用プロセス出力
PSSMWR	直結専用マスク付きプロセス出力
PSLTP	論理・物理変換
PSPTL	物理・論理変換

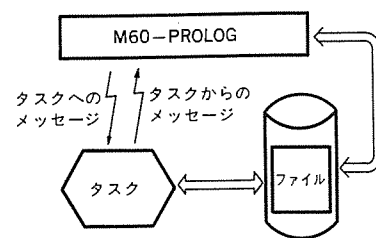
PROLOG プログラムで利用する方法は GSP の場合と同様で、一般形 FCALL FSUB (パラメータの並び)

を用いる。FSUB が SBMS 中の サブルーチン 名であり、パラメータの並びはその サブルーチン の引数に対応する。引数には配列データを持つものがほとんどであるが、《MELCOM 350-60》PROLOG では配列データを複合項として与える。

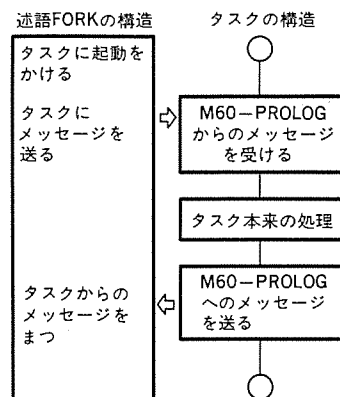
GSP にせよ SBMS にせよ、組込み述語 FCALL により実行される場合、パラメータの個数や形がまちがっているなど処理系が文法エラーを見つけることができるときには、この目標は FAIL する。一方、ランタイムルーチンに制御が移ってからエラーが検知されたときには、このゴールは SUCCESS の形をとるものの、端末にエラーメッセージが表示されるようになっている。

4.3 タスク間通信

上に述べた GSP や SBMS などは既存の FORTRAN パッケージであり、それらは産業用 PROLOG という立場からすれば、汎用的で必ず(須)のライブラリなので、そのまま処理系の中に抱え込ませている。ところでユーザーの立場からすれば、既存の有用なプログラムや FORTRAN で書いた方が、はるかに効率の良いプログラムをも、PROLO



(a)



(b)

図 6. タスク間通信の方式

G プログラムと連結して動かしたい。この要望にこたえるのがタスク間通信機能である。

基本的な考え方は、PROLOG プログラムと別タスクとの間にコミュニケーションデバイスを設定し、それを介してメッセージやデータの交換を行うということである。コミュニケーションデバイスとしてはファイルが便利である。図 6. にその方法を示す。タスクの起動は組込み述語 FORK を用いて行う。次のような形で FORK の引数として起動したいタスク名、タスクに対するメッセージを渡す。

FORK ("タスク名", タスクへのメッセージ, タスクからのメッセージ, エラーコード)

タスクと PROLOG プログラムとの間でやり取りするデータ量が少ない場合 (128 バイト以下) には、コミュニケーションデバイスとして FORK の第 2, 第 3 引数のメッセージ領域を利用できる。PROLOG 処理系は、タスクを起動後、それが終了して FORK の第 3 引数にメッセージがセットされるまでは、次のゴールの実行に移らない。

5. プログラミング支援ツール

PROLOG プログラムを開発する際の支援ツールとして、特に重要なエディタとデバッガについて説明する。

5.1 エディタ

PROLOG プログラムのテキストファイルは、《MELCOM 350-60》の OS のもつエディタにより普通どおり編集できる。ここでは、PROLOG プログラムを実行 (インタプリト) する過程で、エディタを利用する方法を述べる。それには、組込み述語 CALLED をゴールとして実行すればよい。レコード内文字列編集やライン操作 (表示、置換、挿入、削除、移動) は OS のエディタ機能をそのまま利用するが、ファイル操作 (生成、結合、拡張、複写、削除) については、PROLOG 処理系とランタイムルーチンとで機能を実現する。ファイル操作の第 1 ランクのコマンドの一覧を表 2. にあげる。

表 2. エディタコマンド (第1ランク)

機 能	コ マ ン ド 形 式
ファイル生成 エディット	ED, ファイル名 ✓
ファイルの拡張	EXP, ファイル名, 不足したファイルの大きさ ✓
ファイルの結合	CONC, ファイル名 1 [; ライン番号-ライン番号], ファイル名 2 ✓
ファイルの複写	COPY, ファイル名 1, { TO OVER }, ファイル名 2 ✓
ファイルの削除	DEL, ファイル名 ✓
エディタの終了	END ✓

注 ✓ は RETURN キー

? - (USER),
APPEND (<>, *X, *X),
APPEND (<*A/*X>, *Y, <*B/*Z>) : -APPEND (*X, *Y, *Z).

(a)

YES
DEBUG- DEBUGGING,

DEBUG
TRACE
LEASH : CALL EXIT REDO FAIL

YES
DEBUG- APPEND (<1, 2, 3>, <4, 5>, *X).
(1) 0 CALL APPEND (<1, 2, 3>, <4, 5>, *67) ? C.
(2) 1 CALL APPEND (<2, 3>, <4, 5>, *321) ? C.
(3) 2 CALL APPEND (<3>, <4, 5>, *525) ? C.
(4) 3 CALL APPEND (<>, <4, 5>, *729) ? C.
(4) 3 EXIT APPEND (<>, <4, 5>, <4, 5>) ? C.
(3) 2 EXIT APPEND (<3>, <4, 5>, <*731, 4, 5>) ? C.
(2) 1 EXIT APPEND (<2, 3>, <4, 5>, <*527, *731, 4, 5>) ? C.
(1) 0 EXIT APPEND (<1, 2, 3>, <4, 5>, <*323, *527, *731, 4, 5>) ? C.

*X = <*323, *527, *731, 4, 5> ;

(1) 0 REDO APPEND (<1, 2, 3>, <4, 5>, <*323, *527, *731, 4, 5>) ? L.

NO
DEBUG- NODEBUG,
? - NOLOG.

(b)

図 7. デバッグ過程

5.2 デバッガ

PROLOG は FORTRAN のような手続形言語と違って、プログラムをながめただけでは実行の流れを把握しにくいところがある。それは、プログラムの実行がパターンマッチングとバックトラッキングにより進むので、プログラムの書き順に見ていっても実際の実行順序になってはいないためである。したがってデバッグ機能には、実行の流れをよくわかる形で示す手段が必要である。その一つに Box Model と呼ばれるものがある。

これは演えき(譯)過程の実行制御フローを説明するもので、《MELCOM 350-60》PROLOG では組込み述語 TRACE を用いると CALL, EXIT, FAIL, REDO などの制御端子用語が各ゴールの前に

つけられて表示される。TRACE には UNLEASH と LEASH の二つのモードがあり、前者を指定すると表示は無制御に次々に行われ、後者では各時点でユーザーに何をしたいかを聞いてくる。LEASH 時には、トレースの度合いを変えることのできる三つのコマンド (CREEP, LEAP, SKIP) が用意されている。

図 7. の例についてデバッグの概要を説明する。二つのリストを連結する述語 APPEND を図 7. (a) のように定義したが、どうも虫があるらしい。どこか? 現在のデバッグモードを確認するため DE BUGGING と打ち込む。その下に 3 行、モードが表示された。APPEND (<1, 2, 3>, <4, 5>, *X) の動きを、1 ステップずつたどってみる (? の後に CREEP の意味で C と指定)。CALL と EXIT は各々、その前についたネストの深さを表す番号同志が対応する。番号 3 の CALL に対する答が、番号 3 の EXIT になっており、ここまでは正しい動作をしている。ところが番号 2, 1, 0 では、いずれも正しい答になっていない。以上から APPEND の定義の 2 番目の節、特に 3 番目の引数を見直す必要があることが判明する。

6. 期待される応用分野

PROLOG の特長は、述語論理に基づく言語で、再帰的記述、リスト構造処理、非決定的処理、リレーショナルデータベース管理機能などを本質的にそなえている点にある。プログラムとデータの同一化ができるので知識ベースが扱いやすいとか、プログラムの構造が計算機の動作機構を反映しないのでユーザーになじみやすいとか、応用面からも従来言語にない魅力を持っている。

これらの特質から、定理の証明、機械翻訳、自然言語理解などの人工知能研究分野をはじめとして、グラフ理論、代数学、平面幾何などへの記号処理的アプローチ、あるいはコンパイラやインタプリタの記述といった情報処理系自身への応用など各種の試みがなされてきた。

最近では、特に演繹的推論機能を持つ知的データベース検索システムという観点から PROLOG の応用を展開していく動きが活発であり、病気に対する処方コンサルティング、各種設計技術情報支援、旅行案内、買物案内、更に大規模な産業用としては原子力プラント運転支援情報提供システムなど、いわゆる知識工学のエキスパートシステムを構築する試みが国の内外で盛んになろうとしている。

7. む す び

《MELCOM 350-60》PROLOG の言語仕様、制御メカニズム、拡張機能などについて特長を解説した。オンラインリアルタイム向けの工夫として、詳細は省略したが、一部モジュールをアセンブラで記述し、何人かの端末ユーザーに対しては、スタティックキャッシュ上にモジュールを入れて高速実行サービスをできるようにした。この PROLOG が産業応用向けの新世代プログラミング言語として多くのユーザーに愛用されることを願うものである。

1. ま え が き

PROLOG は、知識情報の処理に適した述語論理形 プログラミング 言語である。1970 年代初頭に フランス で着想・開発され、中期に イギリス で実用的処理系に拡張されたが、我が国では第 5 世代 コンピュータプロジェクトにおいて中核的言語に設定されたことにより、大きな注目を浴びることとなった。PROLOG は推論機構など、従来形言語にはない新しい機能を備え、エキスパートシステム や機械翻訳などの高度な知識情報処理用ベース言語として期待が高まっている。

最近になって種々の処理系の開発例が報告されているが、ここで紹介する漢字 PROLOG 処理系は、日本語処理を中心とした知識情報処理のために、漢字・かな機能、及び文字列処理機能を拡張して開発された。処理系はアセンブラによって記述され、《MELCOM-COSMO シリーズ》の UTS/VS のもとで動作する。これに加えて最適化処理、コンパイラ機能などによる妥当な実行速度とプログラム開発支援環境が実現されており、十分実用的な構成となっている。

2. 言 語 機 能

2.1 項の形式

PROLOG のプログラム、データはすべて「項」と呼ばれる記述単位で構成される。漢字 PROLOG では、項の表記に Lisp と同じく「S 式表現」を用いている。

定数は 3 種類あり、整数は 10 進と 16 進表記が許される。アトムはシンボルの列からなる項で、述語名やプログラム中の定数として使われる。文字列は「'」で囲まれたシンボルの列である。変数はアトムの前に「*」を付けた項で、それ自身決まった値を持たず、統合化によって値が割り付けられる。「*」だけからなる「匿名変数」は単に変数でありさえすればよい場所で使われる。複合項は、1 個の述語名、又はファンクタと任意個の引数で構成され、述語呼出し（従来形言語における手続き呼出しに相当）、又は複雑なデータ構造の表現に使われる（図 1.）。

2.2 プログラムの形式

PROLOG のプログラムは、「ホーン節」と呼ばれる述語論理の一形式

項	定 数	{ 整数: 123, -5, & AOB9 アトム: ATOM, 原子 文字列: "STRING", "文字列"
	変 数	: *X, *変数, *
	複合項	: (F A ₁ A ₂ ...A _n) ここで { F: ファンクタ (アトムが充てられる) A _i : 引 数 (項が充てられる)
	リスト	: (L ₁ L ₂ ...L _n) ここで L _i : 任意の項

図 1. 項 の 分 類

に基づき、3 種類の構文をとる。漢字 PROLOG における表現を以下に示す。ただし P, Q_i は上述の述語呼出しであり、ASSERT はプログラム登録用組込み述語である。

- | | |
|---|------|
| ① (ASSERT(P Q ₁ Q ₂ ...Q _m))... | 〔規則〕 |
| ② (ASSERT(P))... | 〔事実〕 |
| ③ Q ₁ Q ₂ ...Q _n ... | 〔質問〕 |

P は頭部、Q₁~Q_i は本体と呼ばれ、それぞれ結論、条件を表す。したがって①は「Q₁~Q_m が正しければ P は正しい」、又は「P を満足するために Q₁~Q_m を満足せよ」という意味になる。②は本体のない形で「P は無条件に正しい」を意味する。③は反対に頭部のない形で「Q₁~Q_n は同時に正しいか」、又は「Q₁~Q_n をすべて満足せよ」を表す。①と②は PROLOG の(副) プログラムを構成する。一方、③は主プログラムに相当するもので、これによってプログラム実行が起動される。

2.3 プログラムの実行

図 2. は日本語文解析のための簡単な文法プログラムであり、①~④

(ASSERT((文章 *文 *END) (NP SUB *文 *X) (VP *X *END)))...	①
(ASSERT((VP *X *END) (NP OBJ *X *Y) (VT *Y *END)))...	②
(ASSERT((VP *X *END) (VI *X *END)))...	③
(ASSERT((NP *マーク *X *Y) (名詞 *マーク *X *XY) (C *マーク *XY *Y)))...	④
(ASSERT((VT (食べる) *END) *END)))...	⑤
(ASSERT((VI (走る) *END) *END)))...	⑥
(ASSERT((名詞 SUB (太郎) *E) *E)))...	⑦
(ASSERT((名詞 OBJ (ケーキ) *E) *E)))...	⑧
(ASSERT((C SUB (は) *E) *E)))...	⑨
(ASSERT((C OBJ (を) *E) *E)))...	⑩

- (文章 *文 ())...	⑪
== SUCCESS ==	
*文 = (太郎 は ケーキ を 食べる)	⑫
== SUCCESS ==	
*文 = (太郎 は 走る)	⑬

図 2. 日本語文法プログラム及び実行例

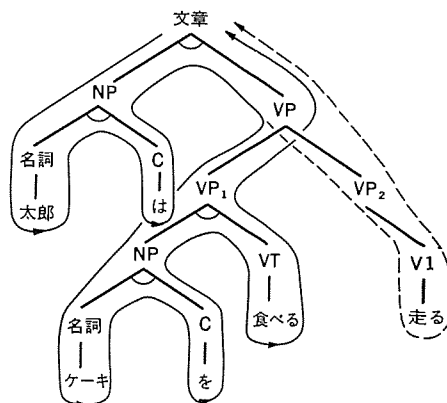


図 3. 「文章」の動き

が規則, ⑤~⑩が事実の形式で表されている。これに⑪のような質問を与えると, 図 3. のような動きをする。つまり⑪の「文章」から始まって①の「文章」へ, 続いて①の「NP」から④の「NP」へと進んでいくが, この過程で行われる形合せ処理, 例えば①と④の「NP」間で SUB=*マーク, *文=*X などとおく処理を「統合化」, 又は「ユニフィケーション」と呼んでいる。統合化は PROLOG の推論機構を支える機能である。

実行が進むと⑫で示される解が得られる。これは図 3. の実線で示された経路の実行に対応する。続いて処理系は「バックトラック」と呼ばれる自動後戻り機能により③から実行を再開し, ⑬で示されるもう一つの解を出力する。これは図 3. の破線で示された経路に対応する。このように条件を満足する解を 1 個だけでなく, 次々と求めていく機能を非決定的機能と呼び, バックトラック がこれを支えている。統合化とバックトラックは, PROLOG における最も基本となる処理機能である。

2. 4 漢字 PROLOG の特長

漢字 PROLOG の言語機能は, エジンバ版 DEC-10 PROLOG に準拠しているが, 機能拡張その他により下記のような特長を持っている。

(1) 漢字・かなデータが使用可能

すべての文字種の内部コードを 2 バイトに統一することにより, 漢字・かな文字が英文字と同等に使用可能とした。コード体系は MEL-COM 漢字コード規格によった。

(2) 文字列機能の実現

DEC-10 PROLOG における文字列は内部的にリスト形式で扱うため効率が悪い。漢字 PROLOG では文字列形データを新設するとともに, 効率的な処理機能を準備した。

(3) コンパイラ機能の実現

プログラムの実行速度向上のために設けた。詳細は 3. 3 節で述べる。

(4) プログラム開発支援環境の充実

一般にプログラム開発を能率良く行うには言語処理系のほかに, 編集系とテストデバッグ系が必要となる。編集系については UTS/VS のもとで稼働しているテキストエディタ「EDIT」を, 漢字 PROLOG 内から呼び出して使用可能とした。テストデバッグ系についてはブレーク, トレース, ステップ実行など, 対話的にプログラムのテストデバッグ作業を行う上で十分な機能を実現した。

(5) 豊富な組み込み述語

従来形言語における組み込み関数と同じく, PROLOG でも組み込み述語が準備され, プログラム中で自由に使うことができる。漢字 PROLOG では約 100 種類の組み込み述語が準備されている。内訳は入出力, 算術, 文字列, コンパイラ, テストデバッグ, その他である。

3. 処理系の構成

3. 1 モジュール構成

処理系は基本的にインタプリタであり, 入力されたプログラムは内部的に 2 進木構造で保持し, 2. 3 節で述べた統合化とバックトラック機能を使って解釈実行する。P モジュールはこの制御の中核である。G モジュールはメモリ管理を担当し, ヒープ領域と 5 本のスタック領域を用いてプログラムと解釈実行用データを管理する。V, B モジュールはテーブル情報, つまり処理系全体で共通的に参照されるデータ, 及び組み込み述語の入口情報を管理する。以上のモジュールが漢字 PROLOG 処理系の核部分であり, 処理効率向上のためにアセンブラ記述となって

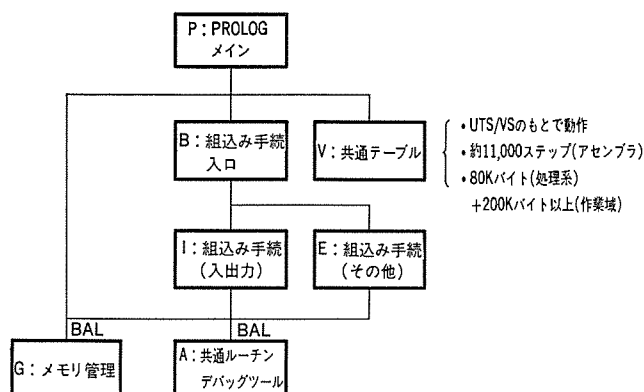


図 4. 処理系のモジュール構成

いる。

一方, I, E モジュールなどの組み込み述語部分は一部を除いて, 漢字 PROLOG 自身で記述されている。A モジュールは更にこれらの子モジュールである(図 4.)。

3. 2 メモリ管理方式

PROLOG の処理系は一般にメモリ消費が激しく, 中間データや実行管理用情報を保持するためのメモリを多く必要とする。したがって如何にメモリの消費を抑制するかが, 実用システムにとって非常に重要なかぎ(鍵)になる。漢字 PROLOG 処理系では, メモリ節約方式として構造共有方式 (Structure Sharing), 及び TRO (Tail Recursion Optimization) を採用している。ヒープ領域 (部分) にいては, ちり集め (Garbage Collection) 機構が用意されている。更にスタックやヒープ領域のオーパフロー時には新たに動的領域を獲得することにより, 実行の延命策を講じている。

3. 2. 1 構造共有方式 (Structure Sharing)

構造を持ったデータを効率的に保持するための方式である。構造データをスケルトン (構造部分) と環境 (値部分) の対形式 (分子という) で持ち, 処理の過程で刻々変化する値の管理は後者にゆだね, 変化しない前者は 1 個で共有することにより節約を図るのである。

図 5. は第 1 引数と第 2 引数で与えられたリストを結合 (APPEND) して第 3 引数に返す, という典型的プログラム例であるが, 解が求まったときのスタック, 及びヒープ領域の状態をその下に示す。(*A, *Z) という構造に対し, 環境の方は再帰呼び出しごと (③, ②, ①の順に呼ばれる) に作成されてスタック領域に積み上げられる

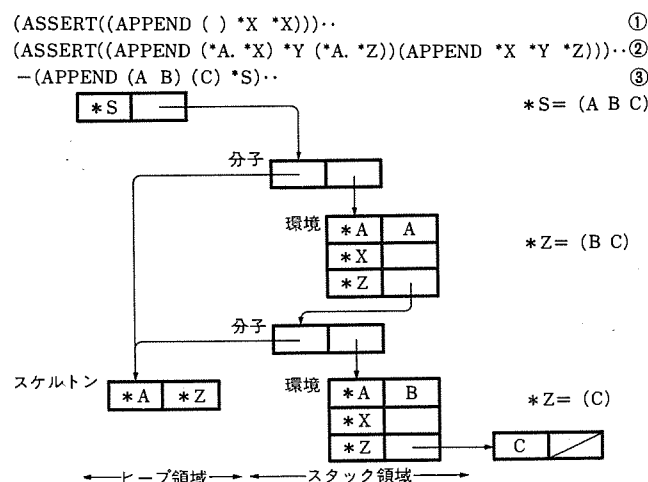
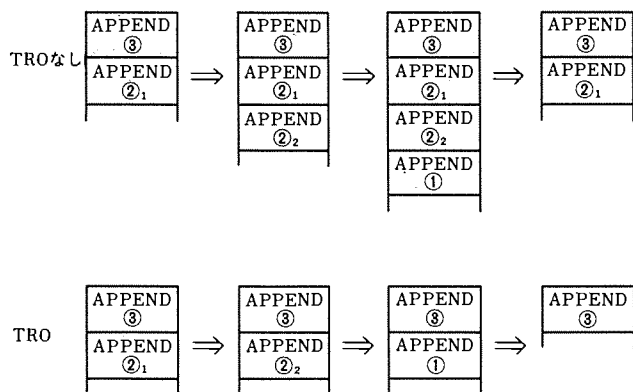


図 5. 構造共有方式によるメモリ管理



最初にAPPEND②が呼ばれたとき 再びAPPEND②が呼ばれたとき APPEND①が呼ばれたとき APPEND①の終了時

図 6. TRO によるスタック領域管理

が、スケルトンの方はヒープ領域に1個作成されるだけである。最終的な解は最初の呼出し時の*Zをたどればよく、変数の値を環境の中の値で順々に置換することにより(A B C)が得られる。

この方式はDEC-10 PROLOG 処理系をはじめ、実用的な処理系の間で広く採用されており、メモリ管理手法の一つの規範となっている。

3.2.2 TRO (Tail Recursion Optimization)

スタック領域の消費を抑制する方式である。PROLOG は非決定的処理をするため、スタック情報は一般にすべての解が求まった後でないと解放できない。しかしある特別な状況下では、正常なスタック解放に先行して解放することが可能であり、処理系がこの状態を検知してメモリの節約を図ることができる。

図 6. は前出のプログラム例 (APPEND) を実行したときのスタック領域の変化を表したものである。この場合、TRO を適用しなければスタック消費量が「第1引数で与えられるリスト長+1」になるのに対し、TRO を適用するとスタック消費量はリスト長とは無関係に「1」となり、劇的な節約効果が得られる。

3.3 コンパイラ機能

PROLOG プログラムをコンパイルすることにより、実行速度を向上させることができる。漢字 PROLOG 処理系ではコンパイラ機能を一群の組込み述語によって実現しており、コンパイル指定、取消し、オブジェクトコード(機械語)のロードセーブ、最適化指定などが可能となっている。

コンパイラは大きく3フェーズに分けられる。フェーズ1はトップレベルコントローラで、指定された手続きの2進木に対して処理用述語を順々に呼び出す。フェーズ2は手続きの定義を基本演算に分解し、インタプリタが実行するであろうような順序でアセンブラコードを生成する。フェーズ3はいわゆるアセンブラであり、フェーズ2の出力を順次機械語に変換する。図 7. はコンパイラの構成及びオブジェクトコードの形式を示したものである。オブジェクトコードのヘッダ部には処理系内の手続きアトムからチェーンが張られており、2進木よりもこちらの方が優先的に実行されるようになっている。

3.4 性能評価

漢字 PROLOG 処理系に対する各種性能測定値を表 1. に示す。一応の世界標準と考えられる DEC-10 PROLOG 処理系に対する性能値がその測定方法とともに公表されているので、これと同じ方法にのっとり測定を行った。実時間比較ではインタプリタモードの場合、DEC-

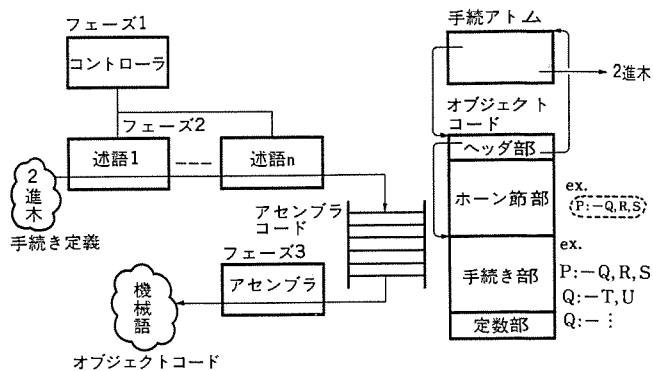


図 7. コンパイラのフェーズ構成及びオブジェクトコードの形式

表 1. 性能測定値 (漢字:《MELCOM-COSMO 900 II》, DEC-10 PROLOG: DEC system 10 KI processor)

処理系	コンパイル時間	オブジェクトサイズ(ワード)	漢字 PROLOG				①/②	
			①DEC-10インタプリタ	②漢字PROLOGインタプリタ	③DEC-10コンパイラ	④漢字PROLOGコンパイラ	①/②	③/④
reverse	6,064	263	1,160	168	53.7	45	6.9	1.2
qsort	10,478	442	1,344	192	75.0	49	7.0	1.5
d-time 10	31,634	971	76.2	13	3.0	6	5.9	0.5
d-divide 10	31,800	971	84.4	14	2.9	6	6.0	0.5
d-log 10	31,798	977	49.2	11	1.9	7	4.5	0.3
d-ops 8	31,887	971	63.7	10	2.2	5	6.4	0.4
serialize	30,656	1,111	602	123	40.2	59	4.9	0.7
query	60,714	2,105	8,888	1,418	185	537	6.3	0.3

10 PROLOG に比べて5〜6倍高速であり、またコンパイルモードの場合はほぼ同等であることが示されている。これはインタプリタの場合、漢字 PROLOG の方はアセンブラ記述で更に最適化処理などにより速度向上に努めたのに対し、DEC-10 PROLOG の方は PROLOG 自身による記述であること、またコンパイラの場合は逆に DEC-10 PROLOG の方がより多くの努力を払った結果と見られる。しかし実使用上は両者とも、ほぼ同等の使用感が得られている。

4. 漢字 PROLOG と日本語処理

PROLOG をベースとしたエキスパートシステムや機械翻訳システムなどの研究開発が盛んになっている。ここでは漢字 PROLOG を使って開発された代表的応用例として「日本語質問応答システム」、「日本語プログラミングシステム」、及び「機械翻訳システム」をとりあげ、PROLOG を使ってどのようなシステムが実現できるのかについて述べる。

4.1 日本語質問応答システム

列車を使った旅行案内を平易な日本語文によって対話的に処理するシステムで、質疑のやりとりが従来のような一問一答形式の「固い」枠組ではなく、必要に応じてシステム側から不足情報を尋ねてきたり、一つの質疑セッションの途中で別の質問を割り込ませることができるなど、より「柔らかい」対話の枠組を持ったものである。

システムは大きく文解析部と問題解決部に分けられ、前者は辞書や文法などの言語知識を使って入力文の解析を行い、文の意味を抽出する。一方、後者はデータベースに蓄えられた対象分野の知識と短期記憶に蓄えられた文脈情報によって、文解析部から渡された意味構造の不備を補い、完全なものにしてから時刻表、料金表などのデータベースを検索する。そして得られた解を文章化してユーザーに返す。知識の表現には人工知能の分野でよく用いられるフレーム形式を使っ

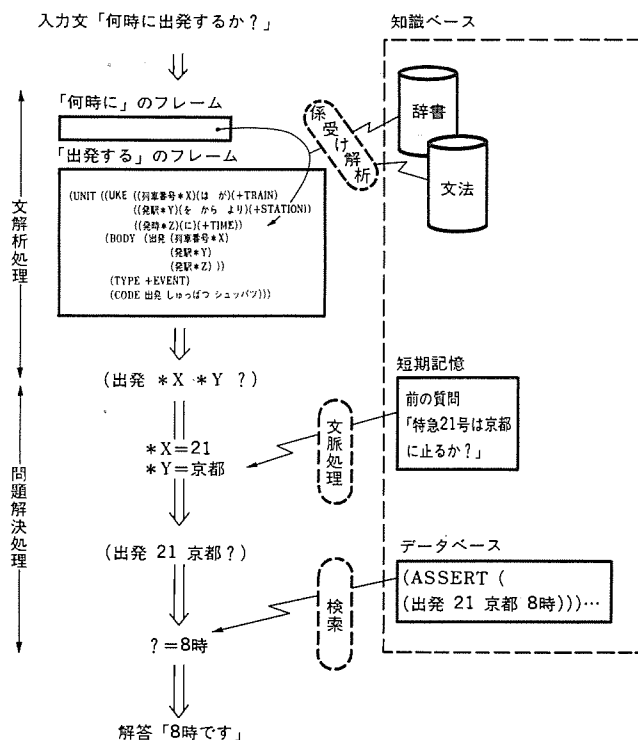


図 8. 日本語質問応答 システム の処理の流れ

```

- (SYSTEM
  入力モードを指定して下さい (KANA/ROMAJI)
  → : KANA.
  ##### 質問をどうぞ！
  → : トッキュウ21ゴウガクノハナシデスカ。
  (特急21号が 着く のは 何時 ですか
  ##### 着駅 ？
  → : トッキュウ21ゴウハナゴヤニマルカ。
  (特急21号は 名古屋に 止るか
  ##### 質問の内容を確認します。
  正しければ (ハイ) , 誤りであれば (イエ) を入力して下さい。
  特急21号は 名古屋に 止るか
  → : ハイ。
  ##### 質問にお答えします e e e はい、そうです。

  ##### 他に質問がありますか？
  あれば (ハイ) , なければ (イエ) を入力して下さい。
  → : イイエ
  ##### では、先ほどの？
  → : ナゴヤデス。
  (名古屋 です)
  ##### わかりました。

  ##### 質問の内容を確認します。
  正しければ (ハイ) , 誤りであれば (イエ) を入力して下さい。
  特急21号が (名古屋へ) 着く のは 何時 ですか
  → : ハイ。
  ##### 時刻 は 8 時 1 分 です

  → : ツギノエキニ ナンジニツクカ。
  (次の 駅に 何時に 着くか
  ##### 前の駅 は 名古屋 ですか？
  e e e 正しければ (ハイ) , 誤りであれば (イエ)
  を入力して下さい。
  → : ハイ
  ##### 列車番号 は 21 ですか？
  e e e 正しければ (ハイ) , 誤りであれば (イエ)
  → : ハイ。
  
```

図 9. 日本語質問応答 システム の対話例

ている。図 8. は「何時に出発するか？」というユーザー質問が処理されていく過程を示したものである。また典型的な対話例を図 9. に示す。

システム全体が漢字 PROLOG で記述され、規模は PROLOG の行数で約 4,000 行である。応答時間も数秒以内で妥当なものとなっている。ここに見られる言語解析や解の探索、マンマシン対話処理など

は PROLOG にとって得意な分野と言える。特に マンマシンインタフェースに自然言語を用いることは、データベースを対象とした自然言語フロントエンドなどへの発展が期待できる。

4. 2 日本語プログラミングシステム

2. 2 節で述べたようにホーン節の頭部は結論を、本体は条件を表す。つまり「本体が正しければ頭部は正しい」という意味になるから、この枠組に従ってプログラムの意味理解を援助するような日本語の構文形式を PROLOG の上に付加すれば、非常に便利であると考えられる。日本語プログラミングシステムはこのような発想に基づいて作成されたもので、図 10. に示されるような日本語構文によって PROLOG プログラムを書くことができる。

処理系の規模は漢字 PROLOG で約 100 行である。PROLOG は構文形式が簡潔である上、意味も論理的に体系付けられているので、このような巧妙な機能が比較的容易に実現できる。この日本語プログラミングは自然言語でプログラムを書こうとする、いわゆる自然言語プログラミングへの第 1 ステップと見ることもでき、OA 分野なども含めた素人向けの簡易なプログラミング言語、その他への応用が考えられる。日本語プログラミングの姿を知る手懸りとして、人工知能分野で有名な 8-Queens 問題に対するプログラム例を図 11. に示す。

4. 3 機械翻訳システム

ここで紹介する翻訳システム「Thalia-1」(サライア-1)は、日本語から英語に文章変換するシステム(第 1 版)であり、規模は漢字 PROLOG で約 4,000 行である。知識表現にはフレーム形式の概念依存構造を用いており、5 種類の辞書と 3 種類の文法を使って翻訳処理が進められる。日本語文は端末、又は外部ファイル経由で入力される。表記には、かな、かな漢字混じり、ローマ字モードが用意されている。

システムは PROLOG の特長を積極的に使用している。例えば翻訳処理の過程で、フレーム構造を使った多くの形合せ操作が必要になるが、これは PROLOG の統合化機能によって処理される。またその過程で発生する多くのあいまい性(構文構造、訳語など)の解決には、PROLOG のバックトラック機能が活用されている。これら他の言

- ① 事 実 : Pである。 : ASSERT(P)...
- ② 規 則 : もし Q_1 であり、 Q_2 であり; : (ASSERT($PQ_1Q_2 \sim Q_n$))...
...ならば、Pである。
- ③ 質 問 : Q_1 であり、 Q_2 であり、 : $Q_1Q_2 \sim Q_n$...
...であるか？

図 10. 日本語プログラミング表現及び等価な PROLOG 表現

```

(定義 交差ししないもの :
  もし *列1 と *距離1 の 和 が *ある列1 であり、
  *列1 と *距離1 の 差 が *ある列2 であり、
  *ある列1 と *列2 が 等しくないものであり、
  *ある列2 が *列2 と 等しくないものであれば、
  *列1 と *列2 が *距離1 の 位置で 交差ししないもの である。)...

(定義 おつからない列 :
  * が ( ) と * の 位置から調べて おつからない列 である。
  及び、
  もし *列1 と *列2 が *距離1 の 位置で 交差ししないもの であり、
  *距離1 と 1 の 和 が *距離2 であり、
  *列1 が *列3 と *距離2 の 位置から調べて おつからない列 である。) ...
  *列1 が (*列2 , *列3) と *距離1 の 位置から調べて おつからない列 である。)...

(定義 衝突しない列 :
  もし *列1 が *列2 と 1 の 位置から調べて おつからない列 ならば、
  *列1 が *列2 と 衝突しない列 である。)...

(定義 条件を満たすもの :
  ( ) が 条件を満たすもの である。
  及び、
  もし *列1 が *列2 と 衝突しない列 であり、
  *列2 が 条件を満たすもの ならば、
  (*列1 , *列2) が 条件を満たすもの である。)...
  
```

図 11. 8-Queens 問題の日本語プログラム

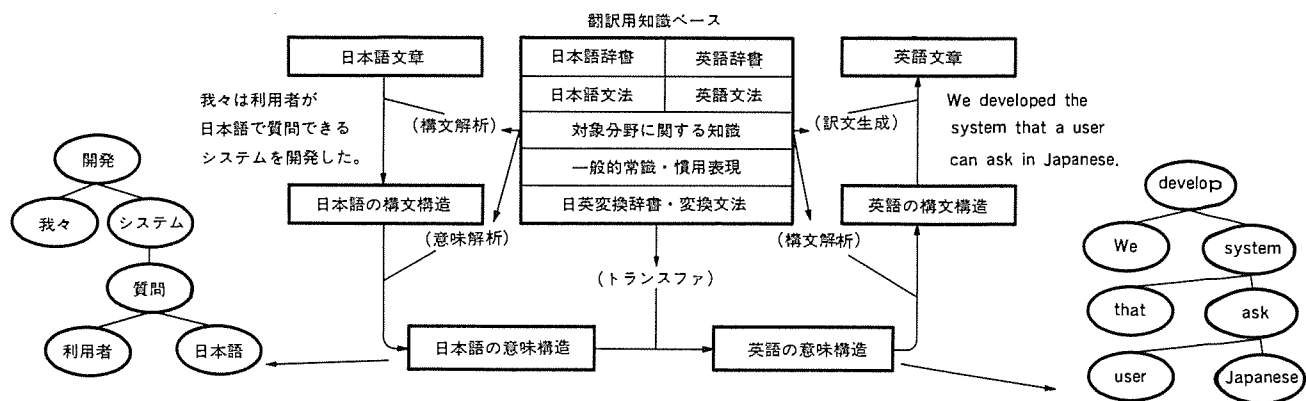


図 12. 日英機械翻訳システム「Thalia-1」の構成

語には見られない優れた言語機能により、システムは見通しよく効率的に構築されており、サイズのにも従来形言語による実現に比べてかなりコンパクトなものとなっている。

機械翻訳は自然言語処理の重要な応用分野であり、長い研究の歴史があるが、最近の知識情報処理の高まりに伴い、PROLOGを翻訳システムの記述言語としてその特長を積極的に利用しようとする試みも活発化してきている。

5. 今後の課題

PROLOGは若い言語であって、様々の優れた機能を持っている一方で、今後につづべき課題も少なくない。その第1は統合化、バックトラックなどの高度な機能である。これらはPROLOG誕生の由来から、現在の計算機アーキテクチャとは無関係に考え出されたため既して効率が悪い。PROLOGに適した新しいアーキテクチャの実現が、並列推論機構なども含めて必要になってくる。第2はデータベース機能である。将来の実応用でシステムは大量のデータや知識が必要とされているが、現在のPROLOGが利用できる領域は主記憶上にとられた内部データベースに過ぎない。この点については、PROLOGと親和性のある関係データベースとのインタフェース機能を実現するか、又は一体化するなどの課題を残している。第3としてマンマシン対話関係では、媒体として図形や画像など、より直観的表現力に富むものを導入してインタフェースの改善を図る方向にあり、このような機能の実現はOA分野への適用拡大などをも考えるとき、重要な課題になりつつある。

以上のような課題がある一方で、PROLOGによるプログラミングの実績を蓄積していくことも大切である。適合分野と言われる知識情報処理分野において、エキスパートシステムや機械翻訳、知的対話システム

など、役に立つシステムの開発を進め、経験と実績を積み重ねることにより、言語の改良・拡張へのフィードバックはもちろん、現時点では判断がつかないPROLOGに対する多くのあいまいもこ(曖昧模糊)とした疑問や問題に対する解を与えることが可能になる。その上で真に適合した分野を探索し、適用していくことが大切である。

一方、経験と実績を蓄積する点では何も新しい分野を指向するだけではなく、既存分野への応用も重要である。例えばPROLOGの持つ概念的表現力の強さ、記述のコンパクト性などは、ソフトウェア仕様記述言語への適性が想起されるし、また知識ベース機能はソフトウェア開発支援システムの構築への適用が期待される。またPROLOGの持つ論理性と、従来形言語の持つ高速演算性を組み合わせて実用的なエキスパートシステムや、意思決定支援システムを構築することは、現時点において最も実現性が高いと考えられ、真剣に取り組む時期にきている。

PROLOGという新しい言語が、真に実用的で有用な言語として定着するためには、このような過程を経て洗練される必要があると考えられる。

6. むすび

以上、漢字PROLOG開発の背景から、漢字PROLOGの言語機能、構成上の特長について、更に漢字PROLOGを使ったいくつかの応用システムについて述べた。漢字PROLOGの概要が分かっていたものだと思う。漢字PROLOGは既に製品化され、第5世代コンピュータプロジェクト関連の研究開発をはじめ、日本語情報処理や機械翻訳などの研究開発に基本ツールとして使用されつつある。今後PROLOGを使った新しい研究開発のため、積極的な活用を期待するものである。

1. ま え が き

プログラムの虫取りや正しさの検証は、依然としてほとんど人の手にゆだねられている。正しさの検証とは、プログラムが与えられた仕様を満たしているかどうかを確かめることであるが、それには一般に多大な手間と時間を要する。そこで、この作業を計算機によってできるだけ自動化しようとする研究が進められている。特に、検証の過程ではプログラムの満たすべき性質を、ある定理として証明することが必要となる。

本文では、当社中央研究所で開発した定理証明システムについて述べる。このシステムは、Lisp などの関数形言語によって書かれたプログラムの満たすべき性質を検証するものである。

2. システムの概要

このシステムでは、次のような式の真偽を証明する。

[equal, [flat 1, x], [flat 2, x, nil]]

flat 1 は二分木 x の葉を一つのリストに並べる関数である。flat 2 は同様のことを別のアルゴリズムで行う関数である。そこで、上の式は同じ仕様に対する異なる二つの関数が、同じ値を与えるものであることを主張している。

2.1 プログラムの構成

システムのプログラムモジュール構成を図 1. に示す。

Main モジュールは、以下のサブモジュールの制御並びにユーザーインタフェースを行う。Definition モジュールは、データタイプと関数の定義を行う。Simplification モジュールは、等式による書換え、データタイプの計算、関数定義の展開などによる式の簡単化を行う。Other Heuristics モジュールは、帰納法適用の前処理として幾つかの書換え規則を試みる。Induction モジュールは、帰納的に定義されたデータタイプや再帰関数に注目して帰納的証明が行われる。

2.2 システムの動作

このシステムにおける証明過程を図 2. によって説明する。以下、証明しようとする式を流体内になぞらえる。

まず、証明しようとする式を (A) に注ぐ。(B) において、式を clause 形式に変換し、(C) に流下する。(C) 内では、様々な書換えを行って clause を簡単化するが、真

偽値にまで書換わったとき、clause は“蒸発”して証明終了となる。(C) 内で“蒸発”しきれず clause がこれ以上書換わらない状況に達したとき、弁 (D) を開き、other heuristics 槽 (E) に流下する。(E) 内で clause が書換わったときは、再び (C) に戻される。(E) 内でもこれ以上書換わらない状況に達したとき、弁 (F) を開き、induction 槽 (G) に流下する。(G) 内では、帰納法により base case と induction step に相当する幾つかの新しい clause が生成され、これらは再び (C) に戻される。

以上の過程で、すべての槽 (C), (E), (G) 内で clause が完全に“蒸発”してしまったとき、最初の式が証明されたことになる。

3. システムの機能詳細

3.1 定義モジュール

このモジュールでは、新しいデータタイプを導入したり、定められた基準に適合する関数定義を公理として受付ける。

(1) データタイプの定義

例えば次のようにして、二分木データタイプ“cons”を定義する。

Add data type “cons” of two arguments with:

```
constructor: cons
recognizer: listp
accessors: car, cdr
```

これにより、次のような公理群がシステムに加えられる。

- (i) [listp, [cons, x 1, x 2]]
- (ii) [equal, [car, [cons, x 1, x 2]], x 1]
- (iii) [equal, [cons, [car, x], [cdr, x]], x]
- (iv) [implies, [listp, x], [lessp, [cdr, x], x]]

(2) 関数定義

再帰的に定義された関数は、再帰呼出しがある measure に関して単調減少する事が証明されるものに限り、受け付けられる。

例えば、関数“append”を次のように定義する。

```
[append, x, y]
= [if, [listp, x],
   [cons, [car, x], [sppend, [cdr, x], y]], y]
```

この定義に対して、データタイプ“cons”の導入の際、システムに加えられた公理 (iv) により、

[append, x, y] > [append, [cdr, x], y]
when [listp, x]

が示されるので、受け付けることができる。

3.2 簡単化モジュール

この定理証明システムの中核を成すモジュールである。このモジュールでは、equivalence を保存するような基本的な書換えを行い、式を可能な限り簡単化する。

(1) if, equal と論理演算子

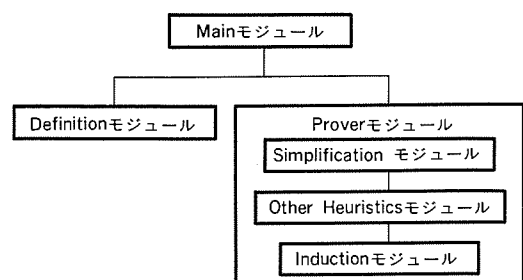


図 1. プログラムモジュール構成

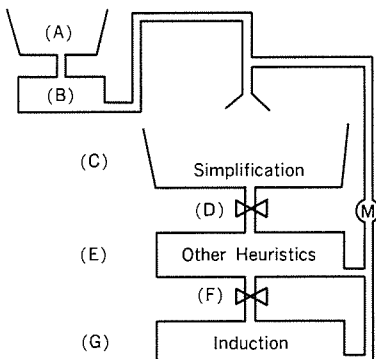


図 2. システム動作説明図

このシステムは次のような真偽値 t, f 、並びに関数 $if, equal$ に関する基本公理をもつ。

公理 $t \neq f$

公理 $x = y \rightarrow [equal, x, y] = t$

公理 $x \neq y \rightarrow [equal, x, y] = f$

公理 $x = f \rightarrow [if, x, y, z] = z$

公理 $x \neq f \rightarrow [if, x, y, z] = y$

論理演算子は次のように if で定義されている。

定義 $[not, P] = [if, P, f, t]$

定義 $[and, P, Q] = [if, P, [if, Q, t, f], f]$

定義 $[or, P, Q] = [if, P, t, [if, Q, t, f]]$

定義 $[implies, P, Q] = [if, P, [if, Q, t, f], t]$

一般に、等式は左辺から右辺への書換え規則として使われる。

(2) type set による簡単化

まずシステムは、式中に含まれる関数に対する type prescription や証明ステップにおける仮定を用いて式の type set を計算し、それらの情報を用いて簡単化を試みる。type set とは、その項が持つ可能性のある値のデータタイプの集合である。その集合の要素は、 t, f 又はデータタイプの recognizer で表される集合 $[recognizer]$ であり、全体集合は universe と表現される。例えば、 $[cons, x, y]$ の type set は “cons” に対する recognizer から $[listp]$ であり、変数 X の type set は universe である。これらの情報を基に項は簡単化される。

今、次の簡単化を考える。

$[equal, [append, x, nil], [add 1, y]]$

“append” の type set はその定義から $listp$ 、又は第二引数の type set であるとわかっているのて、

$[listp, litatom]$

である。また $[add 1, y]$ の type set はデータタイプ定義から、

$[numberp]$

である。こうして $equal$ の両辺の type set の積集合が空なので、この項は f に簡単化される。

(3) 公理と lemma

type set による簡単化に失敗すると、システムは公理や lemma を使った書換えを試みる。これらは、データタイプ定義時に加えられる公理、以前証明された定理、又はユーザーが与える公理や lemma である。項を書換えるのに適当な公理か lemma が見つかったら、それを用いて項は直ちに書換えられる。適当な公理か lemma が複数ある場合は、最初に検索されたものが選ばれる。一般に公理や lemma は次の形をしている。

$[implies, Hyp, [equal, LHS, RHS]]$

これは、“Hyp が証明できるとき、LHS と match する項を RHS に置き換える” という書換え規則として用いられる。今書換えようとする項を、

$[append, [cons, A, B], nil]$

とし、次の lemma があるとする。

$[implies, [plistp, x], [equal, [append, nil], x]]$

ここで、関数 “plistp” はリストの最後の要素が “nil” のとき t を返す関数である。このとき、システムは、項

$[append, [cons, A, B], nil]$

と上の lemma の LHS でるる

$[append, x, nil]$

との match を試み、それに成功するとその環境 $x = [cons, A, B]$ で $Hyp: [plistp, x]$ を証明しようとする。他の条件*1 からこれが証明されたときには、

$[append, [cons, A, B], nil]$

は $[cons, A, B]$

に書換えられる。証明できなければ、別の lemma が試される。

(4) 関数定義の展開

すべての lemma が試されてもお項が書換えられない場合、その項に現れる関数呼出しを定義の body に展開することにより、式は簡単化される。今展開しようとする n 項の関数を、

$[Fn, T_1, \dots, T_n]$

とする。まず、各引数 T_i は、(2)–(4)の一連の書換え規則に従って書換えられる。これらを T'_i とする。 Fn が非再帰関数であるならば、無条件に body で展開される。そして、その body に対して (2)–(4)の一連の処理が再帰的に行われる。

次に Fn が再帰関数の場合を考える。関数 Fn の measured position*2 の引数が explicit value*3 ならば無条件に body で展開される。そうでないならば、まず body が書換えられる。その書換えられた body (body' とする) が下の条件のいずれかを満たせば、

$[Fn, T_1, \dots, T_n]$

は body' に書換えられる。さもなければ、

$[Fn, T'_1, \dots, T'_n]$

が返される。

その条件とは、次の三つである。

(a) body' の再帰呼出しの measured position の項が今簡単化しようとしている式に存在する。すなわち、展開することにより measured position に新しい項が現れるわけではない場合。

(b) body' の再帰呼出しの引数の方が、 $[Fn, T_1, \dots, T_n]$ より多くの explicit value を含んでいる場合。

(c) body' の再帰呼出しの measured position の項の方が、 $[Fn, T_1, \dots, T_n]$ の対応する measured position の引数より単純な項である場合。ここでいう “単純” とは、項に含まれる関数記号の数が少ないことである。

Simplification モジュールでは、式に含まれる各項に対して以上のような一連の操作を行い式の簡単化を試みる。簡単化できなかった場合は、induction を用いて証明を行うことになる。このシステムでは、induction を “上手に” 適用するために式の変形を行う heuristics を持っている。次にその heuristics について述べる。

3.3 ヒューリスティックス

このモジュールは、次の Induction モジュールにおいて帰納法を適用するための前処理として、いくつかの書換え規則を試みる。

(1) Destructor の消去

ここでは、データタイプの導入の際定義された accessor からなる項と constructor からなる項との “トレード” を行う。

例えば下の式において仮定から X がリストであるとわかるので、 X を $[cons, X_1, X_2]$ に $[car, X]$ と $[cdr, X]$ を各々 X_1, X_2 に書換える。この変換によって次の Induction Step において X_1 、 X_2 に関する帰納法を適用することが可能となる。

*1 ここでは、 $B = nil$ 又は $[plistp, B] = t$ の場合。

*2 measured position : 関数呼出しのある引数よりその body の再帰呼出しの対応する引数の方が、ある measure に関して減少しているような引数の位置。

*3 explicit value : t, f , データタイプの bottom object、又は、データタイプ constructor 項でその引数が explicit value であるもの。

```
[impies, [and, [listp, x],
               [equal, [flat 2, [car, x],
                             [flat 2, [cdr, x], y]],
               [append, [flat 1, [car, x]]
                     [flat 2, [cdr, x], y]]],
[equal, [flat 2, [cdr, x], y],
         [append, [flat 1, [cdr, x]], y]],
[equal, [flat 2, [car, x],
               [flat 2, [cdr, x], y]],
         [append, [append, [flat 1, [car, x]],
                           [flat 1, [cdr, x]],
                           y]]]
```

(2) Cross-fertilization

仮定内の equality を結論部に適用する。例えば下の式において、

```
[append, [flat 1, x 1], [flat 2, x 2, y]]
```

を [flat 2, x 1, [flat 2, x 2, y]]

に置換え、その equal 項を取り去る。

```
[implies, [and, [equal, [flat 2, x 1, [flat 2, x 2, y]],
                      [append, [flat 1, x 1],
                                [flat 2, x 2, y]]],
[equal, [flat 2, x 2, y],
         [append, [flat 1, x 2], y]]],
[equal, [flat 2, x 1, [flat 2, x 2, y]]
         [append, [append, [flat 1, x 1],
                           [flat 1, x 2]],
                           y]]]
```

一般に証明しようとする式の仮定に項

```
[equal, x, y]
```

があったとする。ここで、X も Y も explicit value でなく、また

[p, <任意の項>, <y を含む任意の項>]

の形の別の項があるとする。そのとき、P のアジェメントの右側だけ X を Y で置き換え、その等式を元の式から取り除く。この書換え操作を Cross-fertilization と呼んでいる。

(3) 一般化 (Generalization)

ここでは、式の中に 2 箇所以上で現れている項、又は equal の両側に現れている項を変数に置換えて一般化と呼ばれる操作を行う。

例えば下の式で、

```
[flat 1, x 1], [flat 1, x 2]
```

を各々 x 15, x 16 で置換える。

```
[equa 1, [append, [flat 1, x 1]
                  [append, [flat 1, x 2], y]]
         [append, [append, [flat 1, x 1], [flat 1, x 2]]
               y]]]
```

その結果、次のような式が得られるが、これは append の associativity を主張しているものにほかならない。

```
[implies, [and, [listp, [flat 1, x 15]]
              [listp, [flat 2, x 16]]],
[equal, [append, x 15, [append, x 16, y]]
         [apped, [append, x 15, x 16, y]]]
```

ここで flat 1, flat 2 は、常に リストを生成するということがそれぞれの type prescription からわかるので、仮定として、

```
[and, [listp, [flat 1, x 15]], [listp, [flat 2, x 16]]]
```

が付け加えられる。

(4) 不要項の消去

以上の書換えを行った結果、式中に結論部と無関係な仮定となっている項が残る場合がある。ここでは、これらの項を取り除く。ここでいう無関係とは、その項と共通の変数を持つ他の項が式中に存在しない場合をいう。

例えば下の式、

```
[implies, [listp, B]
           [equal, [reverse, [append, z,
                               [cons, A, nil]]]
                   [cons, A, reverse, z]]]]]
```

において仮定の

```
[listp, B]
```

は結論部の equality に含まれない変数 B に関するものなので、取り除く。

3.4 Induction

このモジュールでは、帰納的に定義されたデータオブジェクト並びに再帰関数について、帰納法による証明を行う。ここで用いられるヒューリスティクスは、このシステムの中核となるものである。帰納法の適用は、再帰関数の定義時において有用な情報を生成する処理と、証明時においてこれらの情報を収集、選択して帰納法の形を定める処理とに分けられる。以下各々の処理について述べる。

(1) 再帰関数の定義時の処理

再帰関数に対して、induction の形を示唆する induction template を生成する。

下に前出の関数 flat 1, flat 2 を例として示す。

flat 1 の定義とそれに対する induction template

```
[flat 1, x]
= [if, [listp, x]
      [append, [flat 1, [car, x]],
                [flat 1, [cdr, x]]],
  [cons, x, nil]]]
```

induction template for the term: [flat 1, X]

measured subset: <x>

case: [listp, x]

(1) x → [car, x]

(2) x → [cdr, x]

flat 2 の定義とそれに対する induction template

```
[flat 2, x, y]
= [if, [listp, [x],
            [flat 2, [car x], flat 2, [cdr, x], y]],
  [cons, x, y]]]
```

induction template for the term: [flat 2, X, Y]

measured subset: <x>

case: [listp, x]

(1) x → [car, x], y → [flat 2, [cdr, x], y]

(2) x → [cdr, x], y → y

(2) Induction の実行

証明しようとする式の中に現れているすべての再帰関数について、生成されている induction template を収集する。それらの induction を幾つかの基準により取捨選択して、最終的に一つの induction scheme を構成する。

scheme for the terms: [flat 1, x], [flat 2, x, y]

case: [listp, x]

(1) $x \rightarrow [\text{car}, x], y \rightarrow [\text{flat } 2, [\text{cdr}, x], y]$

(2) $x \rightarrow [\text{cdr}, x], y \rightarrow y$

こうして得られた induction scheme を適用することにより、元の式は、base case と induction step からなる clause の積に書換わる。それらは再び Simplifier に戻される。

Base case:

[implise, [net, [listp, x]]

[equal, [flat 2, x, y],

[append, [flat 1, x, y]]]

Induction Step:

[implies, [and, [listp, x],

[equal, [flat 2, [car, x],

[flat 2, [cdr, x], y]],

[append, [flat 1, [car, x]],

[flat 2, [cdr, x], y]]],

[equal, [flat 2, [cdr, x], y],

[append, [flat 1, [cdr, x], y]]],

[equal, [flat 2, x, y],

[append, [flat 1, x, y]]]

4. システムの評価

システムの評価基準としては、検証可能なプログラムのクラス、検証に要する時間、メモリ容量などがあげられる。このシステムでは、再帰関数については 3.1 節の (1) で述べたような measure と、well-founded な減少関数とが与えられるものに限られる。証明に要する時間は式に含まれる書換え可能な項の数、適用される書換え規則の総数、Induction がかかる回数、Simplification や Induction によって分割される clause の個数などによって決まる。メモリ容量については、証明された式が後の証明のための書換え規則としてシステムに追加される分と、1 回の証明に要する作業領域分とに大別される。

このシステムでは、初等代数における加算、乗算の交換、結合則などが、またリスト処理においては append, flatten, intersection など Lisp 基本関数に関する性質の証明が CPU 時間で数分の order で実行されている。

インプリメンテーション言語としては、PROLOG を採用した。プログラムサイズは、表 1. に示すようになった。PROLOG のソースコードは Lisp など他の言語に比べセマンティカルな密度が高く、その分プログラム開発工数も少ないのが特長と言える。

表 1. プログラムサイズ

file name	source (lines)	interpreted code(words)	compiled code(words)
main	480	5,222	5,693
definition	778	8,462	7,408
simplification	1,860	20,400	20,703
other heuristics	743	7,438	7,214
induction	791	8,812	8,638
utilities	589	5,370	6,256
total	5,242	55,004	55,912

5. む す び

このシステムでは、関数形の Lisp プログラムを対象としたが、PROLOG など、論理形プログラムに対しても同様の検証システムを構成することを検討中である。

なお、この論文は第五世代コンピュータシステムプロジェクトの一貫として、ICOT (第五世代コンピュータ技術開発機構) から受注した外注作業の成果である。ICOT・第二研究室の古川室長には貴重な御指導をいただき、深く感謝の意を表する。

参 考 文 献

- (1) Boyer, R. S. and Moore, J. S.: A Computational Logic, Academic Press (1979)
- (2) Cohen, P. R. and Feigenbaum, E. A. Eds.: The Handbook of Artificial Intelligence, Volume, 3, XII-D., Pitman Books Ltd. (1982) pp. 102-113

付 録

以下、このシステムで証明された定理の幾つかと、その証明に要した CPU 時間を付す。

Theorem COMMUTATIVITY. OF. PLUS

[equal, [plus, x, y], [plus, y, x]]

CPU time: 33.0 second

Theorem ASSOCIATIVITY. OF. TIMES

[equal, [times, [times, x, y], z],

[times, x, [times, y, z]]]

CPU time: 137.0 seconds

Theorem LESSP. PLUS. SUB 1

[not, [lessp, [plus, x, y], [sub 1, y]]]

CPU time: 48.4 seconds

Theorem EQUAL, TIMES. ZERO

[equal, [equal, [times, x, y, 0],

[or, [zerop, x], [zerop, y]]]

CPU time: 33.5 seconds

Theorem REVERSE. REVERSE

[implies, [listp, x],

[equal, [reverse, [revrse, x]], x]]

CPU time: 69.3 seconds

Theorem FLAT 1. FLAT 2

[equal, [flat 2, x, y],

[append, [flat 1, x], y]]

CPU time: 52.1 seconds

Theorem MEMBER. REVERSE

[implies, [member, x, [reverse, y]],

[member, x, y]]

CPU time: 138.8 seconds

Theorem MEMBER. INTERSECT

[implies, [and, [member, A, B],

[member, A, C]],

[member, A, [intersect, B, C]]]

CPU time: 23.6 seconds

1. ま え が き

計算機の利用分野の拡大により、ソフトウェアの生産量が急増している。このため、システム開発費用に占めるソフトウェア開発費用の割合が増大しており、ソフトウェア開発における生産性向上が焦びの急となっている。

以下では、特にソフトウェア開発のための設備環境である PWS (Programming Work Station) システムを中心とし、その開発の歴史、効用を紹介するとともに、PWS の適用システムにつき紹介し、今後の動向についても言及する。

1.1 ハードウェア生産設備とソフトウェア生産設備の対比

まず、計算機分野におけるハードウェア製品とソフトウェア製品について、その生産工程を図1.に対比してみた。図1.からそれぞれの製作フェーズごとに、工程を対応できることが分かる。ただし、ハードウェアでは、既に部品化、治工具(ツール)の標準化、高度化が顕著であり、ロボットを導入などCAM/CATによる革新が行われている。そして設計のフェーズにおいても、CADのシステムが浸透し、ハードウェアのコスト低減、合理化に多大な効果をあげている。

一方、ソフトウェア開発においては、紙と鉛筆の手工業から、まだまだ脱皮できていないのが実態であり、開発ツール、生産設備の充実が求められている。

このような背景から、特にソフトウェアの製造部分の合理化、生産効率向上に着目し考案された開発システムとして、PWSの考え方の基本となるPWB(Programmer's Work Bench)が提案されてきた。そして現在は、設計の効率化、システム試験の充実など、その適用範囲も広がり、ソフトウェアツールデータベース、設計支援システムなどの開発システムと併せ、総合生産システムとして、ソフトウェアの生産性向上、信頼性向上に効果をあげてきている。

1.2 PWS の概念の由来

PWS システムの概念の由来は、米国のベル研究所にて1970年代に、

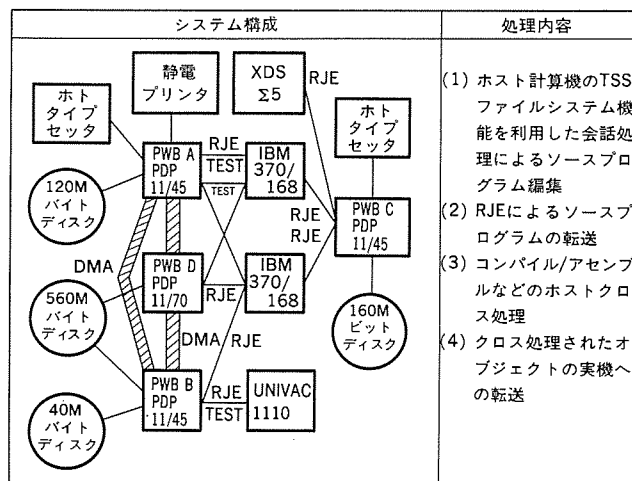


図 2. ベル研究所における PWB システム

ミニコンピュータ (PDP-11) 上で、UNIX をベースとして作られた PWB と呼ばれる ソフトウェア 開発 システム がある。PWB のシステムは、一つの製造システムであり、会話形式で100以上の開発者を同時処理することができる、大規模プロジェクトサポート専用のワークベンチとして位置付けられ、ベル研究所においては、IBM, UNIVAC などのホスト 計算機のためのアプリケーションソフトウェアを開発するために利用された。

そして現在ではマイクロプロセッサ用ソフトウェアを開発するためのホスト機として、各方面で利用されており、経済性、方式の有効性など蓄積された経験による強力なサポートがなされている。図 2. に PW B システムの構成と環境を示す。

2. PWS の特長

ここで紹介する PWS システムは、大形計算機《MELCOM-COSMO 900 II》の上に構築している。図 3. に PWS システムの構成と機能

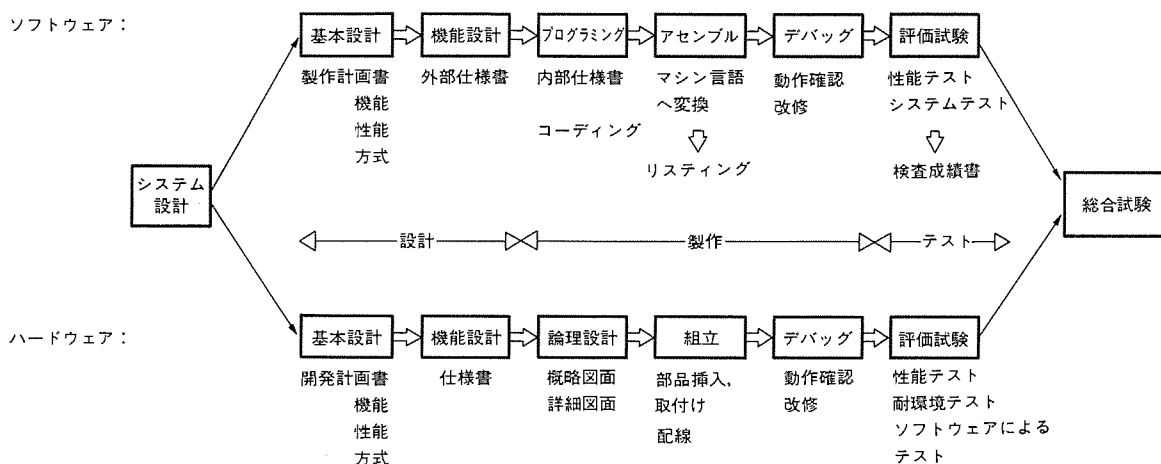


図 1. ソフトウェアとハードウェアの製作過程の対比

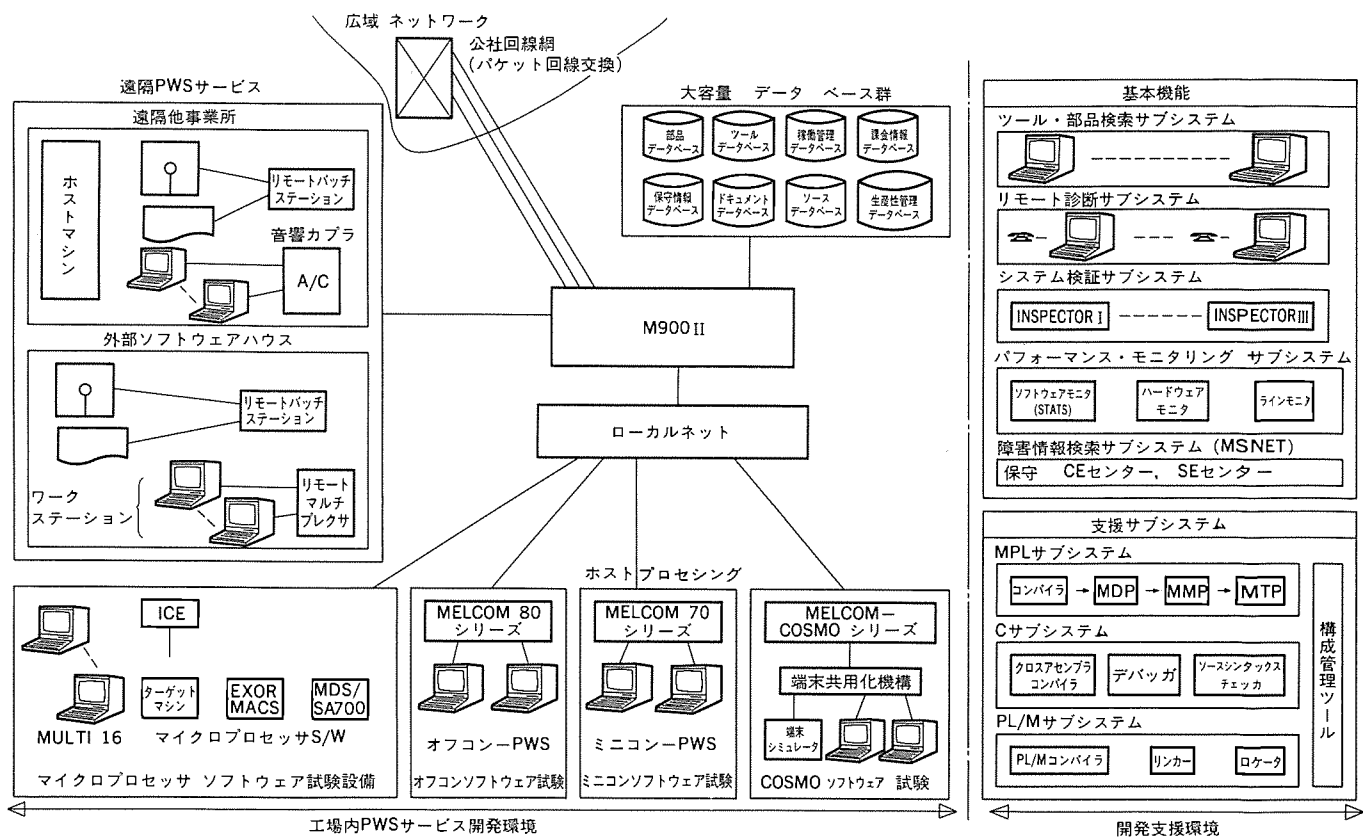


図 3. PWS システムの構成と環境

を示す。このシステムは、次の特長をもっており、ソフトウェア開発者が自席で一連の開発作業を行えるシステムになっている。

- (1) 各種計算機をローカルネットで統合するとともに他事業所、ソフトウェアハウスへの広域・遠隔ネットワークをも形成している。
- (2) 《MELCOM-COSMO シリーズ》のオペレーティングシステム UTS/VS(Universal Total Processing System/Virtual Storage)の基本機能をベースに、ツールデータベースシステム、各種計算機用ソフトウェア開発支援サブシステムなどが体系的に整備され、ネットワーク内の各計算機が有機的に活用できるようなシステム構成となっている。
- (3) ソフトウェア生産用計算機とターゲット・試験用計算機を同一ネットワークに組み込んだことにより、設計から製造・試験までの一環生産が可能となっている。
- (4) マイクロプロセッサ、ミニコンピュータ、オフィスコンピュータなどの低位の計算機のソフトウェア開発用として、クロスアセンブラ、クロスコンパイラ、クロスリンク、実行シミュレータなどのクロスソフトを整備し、大形計算機の高性能なリソースや機能を十分活用した、ホストプロセッシング処理が可能なシステムとなっている。

3. PWS システムの構成要素

3.1 PWS のハードウェア構成

ここで紹介する PWS システムは、会話形処理を主体としてソフトウェア開発を行うための、複合計算機システム環境であり、次の構成要素から構成されている。

- (a) 作業台 : ワークステーション
- (b) 道具 : ソフトウェアツール
- (c) 部品 : ソフトウェアパーツ
- (d) 作業標準 : 利用ルール(ハンドブック)、システム稼働ルール

各構成要素の内容を次に紹介する。

(1) 作業台(ワークステーション)

現在、簡易端末 M 2311 A、日本語ワークステーション M 4378、パーソナルコンピュータ《MULTI 8》、《MULTI 16》などが、ワークステーションの主流となっている。今後は、ソフトウェア CAD の一環として、設計支援、プログラム図作成支援用の図形端末を、これらのワークステーションに加えた形で強化してゆくことにしている。

(2) 道具(ソフトウェアツール)

スクリーンエディタ、コンパイラ、デバッガ、テストベッド、リンク、品質統計ツール、工程管理ツールなどがこれにあたる。これらのツールは後述するツールデータベースにより、一元管理されており、ソフトウェア開発者は、ワークステーションを通して必要に応じて利用できる状態となっている。ハードウェア分野で行われているものと比較すると、歴史も浅く、まだまだ未整備の状況であるが、PWS システムとして、またソフトウェアの工業化を進める上で重要な構成要素であり、今後共拡充してゆく必要がある。

特に次のツールを中心に、ソフトウェアの CAD 化、信頼性向上、システム評価の強化などソフトウェア製作全般にわたる支援ツールへ向け整備することとしている。

- (a) システム仕様解析ツール
- (b) 設計仕様解析ツール
- (c) テストケース生成ツール
- (d) 複雑度評価ツール
- (e) コマンドインタプリタ
- (f) プログラミングライブラリ

(3) 部品(ソフトウェアパーツ)

プログラムの制御や処理を含む構造の類似性に着目し、部品化を進め

ている。表 1. にソフトウェア 部品の定義と分類を示す。小さなものは 0 ステップ 程度のものから、100 ステップ 以上のものまであり、部品化のねらいを以下に定め、ツール 同様、部品 データベース として一元化している。

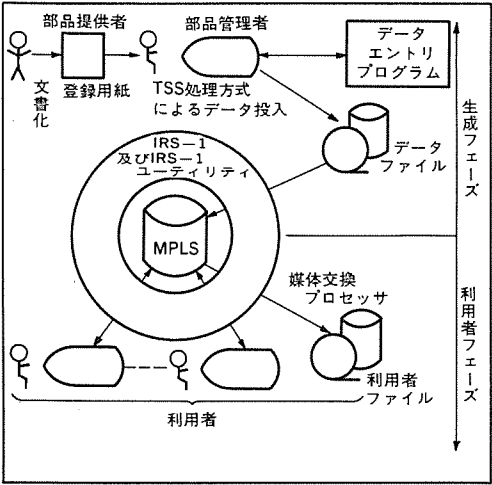
- (a) 重複投資の排除
- (b) 流通化促進

表 1. ソフトウェア 部品の定義と分類

ソフトウェア部品の定義	
定 義	具 体 的 事 項
汎用性がある	繰り返し利用できること
内部ロジックを意識しないでそのまま使用できる	部品の解読なしで利用できること
信頼性が保証されている	虫がないこと
部品はソース又はロードモジュールの形態である	計算機に直接入力できること
標準化された文書（仕様書）がある	統一的な登録用紙に記入されていること
入出力データが明確である	閉じた論理構造をもつこと

ソフトウェア部品の分類	
機 能	内 容
調 査 計 画	新規開発のための前準備作業（他機種の情報収集）
設 計	設計の図形表現，プログラムのアルゴリズムのチェック
製 造	文字列処理，表操作，交換，編集，入出力基本処理
デ バ ッ グ	トレース，メモリダンプ，レジスタダンプ，データの比較
保 守	ロードモジュール，オブジェクトモジュールの修正など

MPLSの構成



検索出力例

DATE	=820906
STEPS	=000045
OPERATION	CALL APLSRT2 (ARG1, ARG2, ARG3)
NOTE	整列の対象となる配列は、主記憶上の連続領域に存在すること。
TYPE	SOURCE
FILE	APLSRT2
LANGUAGE	APL
TITLE	APLツール (APLSRT2)
FACILITY	SHELL法により、与えられた文字列配列を 整列するサブルーチン。 引数は、(1) 配列第1要素、(2) 配列要素数、 (3) 配列要素バイト長。 配列要素は、nバイトの文字列データとして処理される。

- (c) 保守作業の効率化
- (d) 作業標準
- (a) PWS ハンドブック などの利用者向け マニュアル を準備し、各開発作業への サービス 安定化と初心者でもすぐ利用可能となるよう配慮をしている。
- (b) ユーザーサービスレベル の確保 (TAT, レスポンスタイム) の規準設定により、各利用者が一定の稼働環境下でサービスを受けられるよう、次の制限を設定している。

- (i) 各ワークステーション へ割り当てる クォンタム 値
- (ii) 各ワークステーション へ割り当てる CPU 値
- (iii) 各ワークステーション へ割り当てる メモリストレージ
- (iv) 各ワークステーション へ割り当てる パーソナルディスク 容量

3. 2 PWS システムの基本機能

次に PWS 基本機能について紹介する。

3. 2. 1 システム機能

当社 UTS/VS のもつ次の機能を活用している。

(1) ネットワーク 機能

MNA(マルチシェア ネットワーク アーキテクチャ) に準拠した ネットワーク の形成により、各種計算機、端末群を有機的に結合し、相互の資源の共有化、計算機間 (MELCOM-COSMO TO ミニコン, オフコン, マイクロプロセッサ など) における、アップロード/ダウンロード 処理が円滑に行える統一したシステムを構築している。また、ソフトウェア 生産量に対応した、システムアップグレード が容易となっている。

(2) コマンド 機能

ストリング 処理 (EDIT), ファイルオペレーション (PCL) から、コンパイル/ア

ツール登録用紙

ID	ASM	FORTTRAN	DATE	198
記述言語	COBOL	MPL C	主メモリ占有	Words
	PL/I	その他 ()	ソースステップ	
ツール名称			登録番号	PT
機能			クラスコード	
登録者	(所属)		(氏名)	
ツール使用機種	700 80 70 2350	MULTI 16	ツール使用OS	UTS/VS DPS III UOS TCOS
修正履歴	(旧版)		目的機種	CP/M その他 ()
			目的OS	
			目的分野	設計・製作・検査・保守・管理
流用元			ツール形態	言語プロセッサ リンカ シミュレータ ユーティリティ ライブラリ シンボリックデータ その他 ()
操作モード	TSS BATCH TRANSACTION		REAL-TIME	その他 ()
JCL (呼出方法)			注意事項	
			使用方法	
関連資料				
提供状態	SOURCE	ROM	LM	媒体 MT CARD DISK FDD
ファイル名			アカウント	

該当ツールをソフトウェア開発センタに導入したい場合は申請書による手続きが必要です。

図 4. MPLS システム の構成と検索出力例

センブル (COBOL, FORT, PL/I, APL など), リンケージ (LYNX), デバッグ (FDP, DELTA, FDP), 実行 (START) と一連のプログラミングコマンドによる, プログラミング環境を活用している。またワークステーションからのバッチジョブ (BATCH) への挿入, 状態チェック (JOB), 結果の確認 (DEMAND), JCL のカタログ機能, ジョブ処理間の制御 (CL) を行う機能も併せて活用している。

3. 2. 2 ツール・部品データベース

PWS のサブシステムとして, ソフトウェアのツール化, 部品化は生産効率を上げる有効な手段として欠くことのできないものである。このため MPLS (MELCOM Program Library System) を, 支援サブシステムとして開発した。MPLS は, UTS/VS のもとで使用できる, オンライン汎用情報検索システム IRS-1 (Information Retrieval System) を核として構築している。

特長は, 日本語で検索の入出力ができ, かつデータベースシステムを利用しているため, 多量かつ多様なツール・部品を利用目的に合わせて素早く検索することができる。MPLS システムの構成と検索出力例は図 4. のとおりである。

3. 2. 3 操作性

ワークステーションから, 次の機能をサポートしており, 操作性を向上させている。

(1) ワンタッチオペレーション

使用頻度の高いコマンドをあらかじめ, ファンクションキーに割り付けておくことにより, ワンタッチの指示が可能となっている。

(2) デマンド出力

ワークステーションから, ラインプリンタへ出力されるイメージをそのまま見ることができ, 実行結果をチェックした後に, ラインプリンタ装置へ出力したり, 削除したりすることができ, 不要なプリントを防止できる。

(3) 会話形 オンラインデバッグ

会話形 オンラインデバッグにより, プログラム開発が簡単に行える。

(4) 周辺装置のアクセス

ワークステーションからでも直接, ディスク, テープ及びラインプリンタなどの周辺装置を使用できる。

3. 2. 4 信頼性

PWS システムの信頼性確保のため, リモート診断ファシリティを活用し, 遠隔の保守センターから, 常時システム稼働状況を監視するとともに, オペレータ集中監視室にリモートコンソールを常設し, 予防保守をはじめとしたシステム安定稼働実現を図っている。また, システム構成変更時のシステム検証システム (INSPECTER) を開発し活用している。

システム検証システムは次のように, 基本部の検証から実稼働までが検証可能なシステムとなっている。

(1) INSPECTER-I

OS の基本的な機能, 言語プロセッサ, システムユーティリティ (UTS/VS) の動作検証システム

(2) INSPECTER-II

基本的なバッチジョブとネットワークシミュレータを組み合わせた検証システム

(3) INSPECTER-III

実稼働をシミュレートする総合検証システム

3. 2. 5 モニタリング (パフォーマンスアナライズシステム)

システム稼働を定量的・分析的に評価するツールとして, UTS/VS が提供しているソフトウェアモニタ (STATS) と市販のハードウェアモニタを活用している。両モニタのもつ特長をいかし, かつ相互に補完するように, 次の使い分けをしている。

(1) ソフトウェアモニタ

ユーザタスク, OS の各モジュールプロセッサの Activity, 使用頻度など。

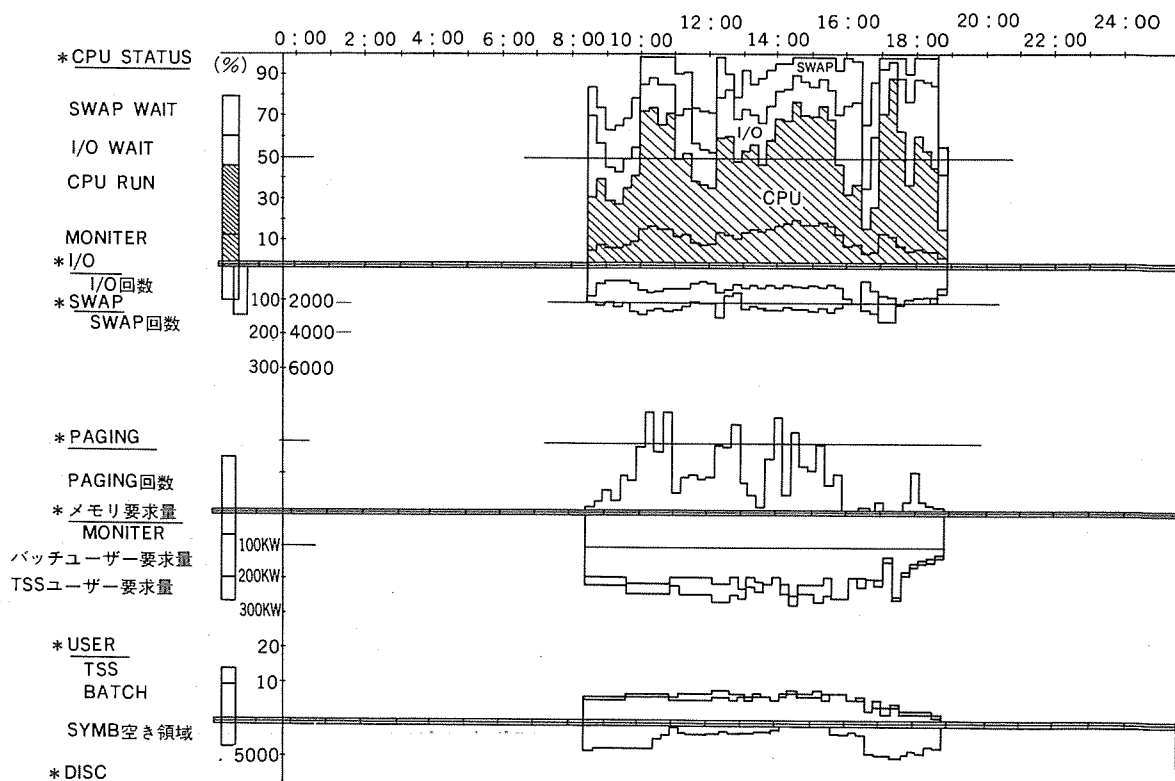


図 5. PWS システムの性能解析例

表 2. MPL 支援 ツールの処理概要と特徴

MDP	MMP	MTP
<p>(1) オンラインモード及びバッチモードで使用できる。</p> <p>(2) MPL原始プログラムで書かれたデータ名や手続名を使ってプログラムの実行を制御し、デバッグ作業ができる。</p> <p>(3) 必要なデータの内容を参照/変更することができる。</p> <p>具体的機能としては</p> <p>(1)区切り点（ブレイクポイント）の設定</p> <p>(2)実行された文・手続名のトレース</p> <p>(3)実行の割込み、再開</p>	<p>(1) オンラインモード及びバッチモードで使用できる。</p> <p>(2) オンラインモードでは、コマンドに対して入力可能なオペランドのリストが表示されるので、利用者は、会話形式でMMPを動作させることができる。</p> <p>性能測定データとしては</p> <p>(1)実行回数（文、手続き、プログラムの三つのレベルがある）</p> <p>(2)データのアクセス回数などがある。</p>	<p>(1) オンラインモード及びバッチモードで使用できる。</p> <p>(2) 単体テストを支援できる。</p> <p>(3) テスト作業の標準化と自動化が図れ、テスト実施者の手作業を軽減し、テスト作業の効率化信頼性の向上が図れる。</p>

(2) ハードウェアモニタ

CPU 利用率、Disk/Channel Activity、キャッシュのヒット率など。

これらの情報を動的にフィードバックし、システム稼働のパラメータの変更を行うシステムを現在開発中である。図 5. に PWS システムの性能解析例を示す。

3. 3 PWS 支援サブシステム

次に 3. 2 節で述べた基本機能を核に、ソフトウェア開発の生産性向上、信頼性向上を図る目的で開発したシステム記述言語をベースとした、MPL サブシステム（大形計算機用）、C サブシステム（大形機、小形機、マイコン用）、PL/M サブシステム（マイコン用）の各支援サブシステムについて述べる。

3. 3. 1 MPL サブシステム

MPL(MELCOM Programming Language) は、当社大形計算機《MELCOM-COSMO シリーズ》用に開発した PL/I 形のシステム記述言語であり、以下の MPL 支援ツールから構成されている。表 2. にこれらの MPL 支援ツールの概要と特長をまとめて示す。

(1) MDP(MPL Debugging Package)

MPL 言語で作成したソフトウェアのデバッグを支援する。

(2) MMP(MPL Monitoring Package)

MPL 言語で作成したソフトウェアの性能評価を支援する。

(3) MTP(MPL Testing Package)

MPL 言語で作成したソフトウェアのテストプログラム作成を支援する。

3. 3. 2 C サブシステム

C サブシステムは、ソース/ターゲットの概念に基づき構築されており、PWS を開発システムとし、ターゲットシステムとしては、COSMO 自身、小形機、マイコンシステムのいずれでも可能である。図 6. にソース/ター

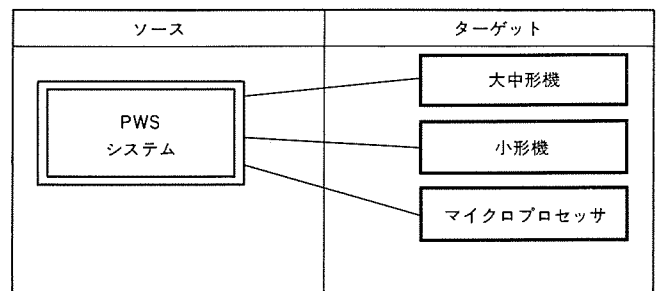


図 6. ソース/ターゲットの概念

表 3. C サブシステム 支援ツールとその内容

ツール名称	内 容
1 Cクロスコンパイラ アセンブラ リンカ	ソースプログラムを入力し、8086用のアセンブラソースを出力する。このソースをクロスアセンブラ、クロスリンカを通すことにより、8086上のCP/Mで実行可能なロードモジュールに変換する。
2 ソースシンタックス チェッカ	C言語で書かれたソースプログラムを対象として、文法上のエラーチェックを行うほか、未定義変数の使用や冗長なコーディングなどにつき指摘する。
3 デバッグパッケージ	会話形シンボリックデバッガができ、関数トレースやステップショットダンプ機能が準備されている。
4 ダウンローディング ユーティリティ	PWSでクロスコンパイル、リンクされたROMやロードモジュールをターゲットシステムに転送する。ソースファイルとしてグラフィックコードのほか、漢字コードや英小文字を含むことができ、バイナリも転送可能である。

ゲットの概念を示す。表 3. に C サブシステムの支援ツールとその内容をまとめて示す。

3. 3. 3 PL/M サブシステム

計算機の高性能化、多機能ニーズに対応してマイクロプロセッサ応用製品が、各計算機関連コンポーネントに組み込まれている。これに伴って、マイクロプロセッサソフトウェアの生産量が急増している。このためのソフトウェア生産の統一を図り合理化を図る目的で、PL/Mをベースとした複合生産システム(MSP)を開発した。このシステムは、《MELCOM-COSMO シリーズ》計算機をホストマシンとし、その高速性、大容量性を活用したシステムとなっており、プログラム言語処理機能のほかに、試験、品質評価及び生産管理のための支援ツール群を一貫システムとしてもっている。

これらはすべて、ホストマシン上でクロス処理されるが、ソフトウェアテストの効率化のために、ホスト機上のシミュレータのみでなく、ターゲットCPUでリアルタイム実行もできるようにしている。デバッグ機、ICE(In Circuit Emulator)も利用できる。図7.にMSPの概念を示す。表4.にはMSPの構成をまとめて示す。

4. む す び

今後のPWSの動向については、従来の既存の計算機環境をそのまま利用した形での生産設備から、ソフトウェア生産に適した設備や、計算機環境を作りあげてゆく動きが、ますます強くなってゆくことが考えられる。

具体的には、

- (1) ソフトウェア開発専用オペレーティングシステムを使用した開発環境
- (2) 端末のインテリジェント化、多機能化とLAN(ローカルエリアネットワーク)の融合したシステムの構築
- (3) ソフトウェアツール、部品化の標準化、流用化の強化
- (4) 設計から総合試験までの、各開発支援サブシステム環境を含めた一貫システム化

などが挙げられる。また、各種計算機間のネットワークの形成については、LANをベースに、図8.のような設備構成へと発展してゆくものと予想される。

ここで紹介したシステムでの貴重な体験をもとに、新しい課題に向け今後共努力してゆく所存である。

参 考 文 献

- (1) Hünk. H. editor: "Software engineering environment"

- (4) "ソフトウェア・ツール", bit 臨時増刊(昭56-4)

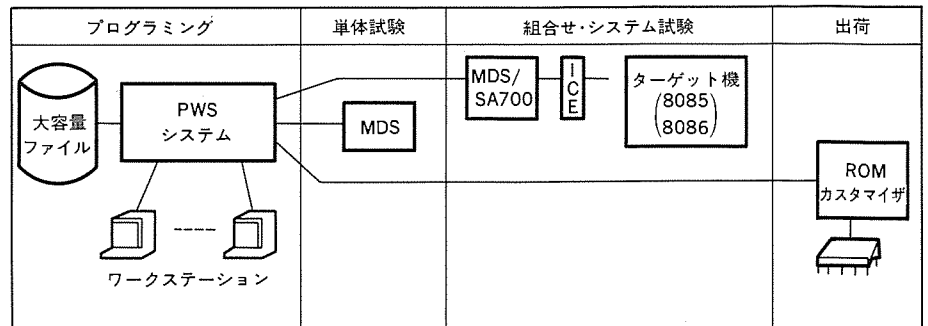


図 7. MSP の概念

表 4. MSP の構成

区分	言語処理	テストデバッグ支援	生産支援	ユーティリティ
機能	<ul style="list-style-type: none"> ・アセンブラ ・PL/M コンパイラ ・Cコンパイラ ・リンカ ・ロケータ 	<ul style="list-style-type: none"> ・マシンシミュレータ ・ソースレベルデバッグ ・テスト環境定義 ・網羅性、性能評価 ・デバッグ機 	<ul style="list-style-type: none"> ・プログラム管理 ・リソース管理 	<ul style="list-style-type: none"> ・ROMフォーマッタ ・ソースエディタ ・ホストデバッグ機接続 ・ホストMDS接続

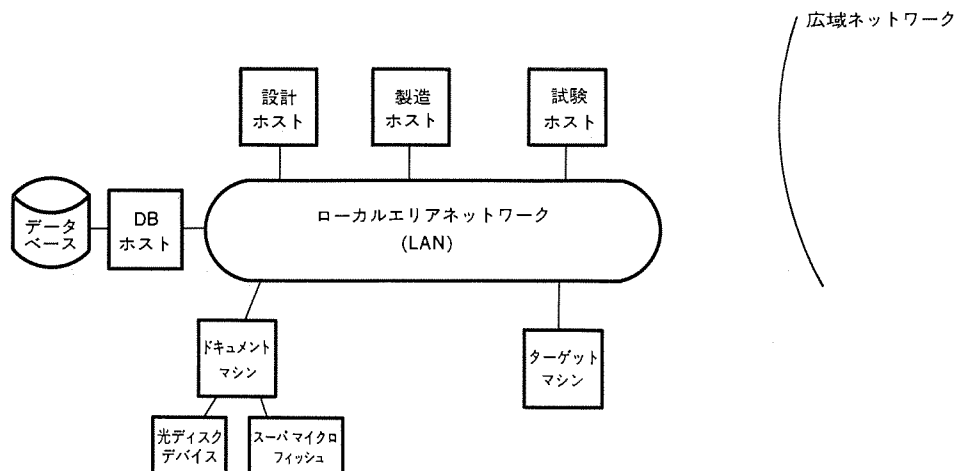


図 8. 将来のソフトウェア生産設備環境

North Holland Publishing Co. (1981)

- (2) ソフトウェア生産技術の現状と将来, 電子通信学会誌(昭58-4)
- (3) T. A. Dolotta, J. R. Mashey (Bell Laboratories): "An Introduction to the Programmer's Workbench.", EDP Analyzer.

同上文献の訳: 五十嵐真訳; bit 臨時増刊, p. 414 (昭57-2)

マイクロプロセッサのソフトウェア開発支援システム

藤掛 遼*・西田親生*・福原直巳*・長井秀憲**

1. ま え が き

数々の産業分野においてマイクロプロセッサの利用が急速に拡大している。これに伴って、生産するソフトウェアの量も膨大なものとなってきており、ソフトウェア開発の生産性向上、信頼性向上が強く求められている。

製品開発のサイクルに対応して、開発支援、試験支援、生産支援、管理支援を統合化したソフトウェア開発支援システムは、そのための一つの解である。ソフトウェア開発支援システムは、プログラミングワークステーション (PWS)、システム記述言語、エンジニアリングデータベース、ソフトウェアツールなど、ソフトウェア工学の成果を集大成することにより、使いやすく、かつ品質の高いプログラムを効率良く開発できるようにすることをねらっている。

このようなソフトウェア開発支援システムとして、システム記述言語“C”をベースとした8086マイクロプロセッサ用ソフトウェア開発支援システムを構築した。このシステムは《MELCOM-COSMO 900II》上でホスト処理するクロスシステムとして構築されている(図1.)。

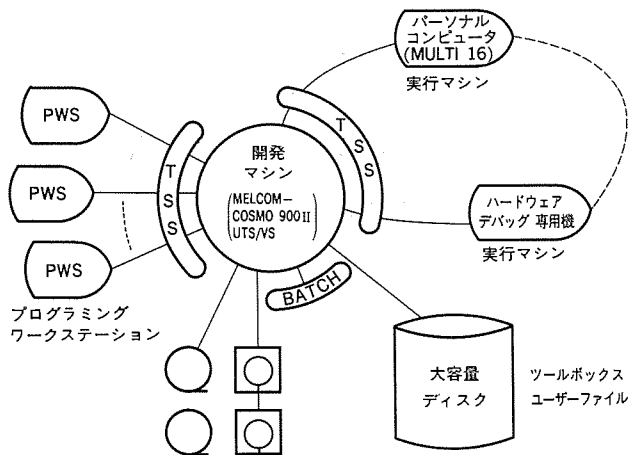


図1. システム環境

2. クロスシステムの特長

クロスシステムは、開発マシンと実行マシンを分離して、開発マシン上でプログラムの完成度を上げて行く、ソース/ターゲットコンセプトに基づいている。このシステムでは、汎用コンピュータ《MELCOM-COSMO 900II》を開発マシンとし、パーソナルコンピュータなどのマイクロプロセッサ応用製品やマイクロプロセッサ応用ハードウェアデバッグ専用機を、実行マシンとして位置付けている。

このシステムは、C言語によるソースプログラムの作成、コンパイル、アセンブル、リンク及びテストまでを一貫して開発マシン上でホスト処理することができる。すなわち、《MELCOM-COSMO 900II》のプログラミングワークステーションで、会話形にC言語のソースプログラムを作成した後、《MELCOM-COSMO 900II》上のCコンパイラにより、コンパイル

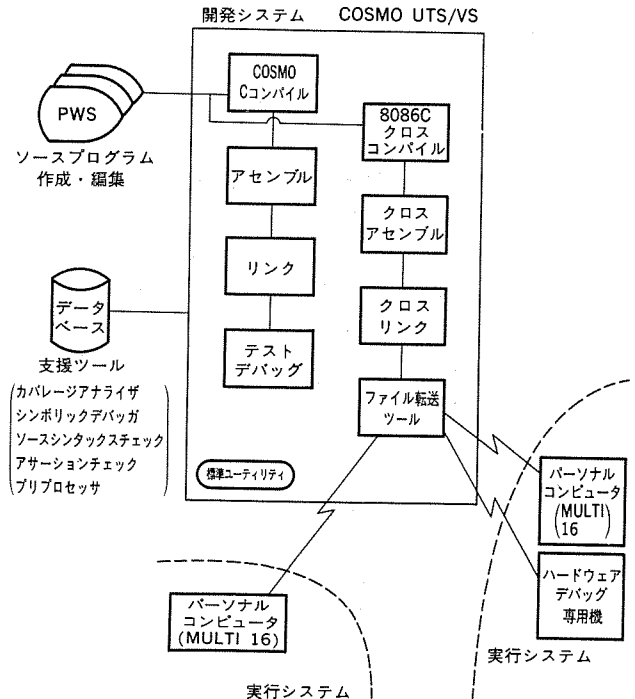


図2. クロスシステムの処理環境

～リンクし《MELCOM-COSMO 900II》上での実行モジュールを作る。この実行モジュールを、シンボリックデバッガ、カバレッジアナライザなどテスト支援ツールの下で実行させることによりテストし、プログラムの完成度をできる限り高める。これらの作業を支援するために、使い方の手引きをファイル化したガイダンス機能やコマンドメニューを示すヘルプ機能なども準備している。

開発マシンである《MELCOM-COSMO 900II》上でのテストが終わったら、次にクロスコンパイラ、クロスアセンブラ、クロスリンカにより、8086マイクロプロセッサ用の実行モジュールを作り、通信回線でこの実行モジュールのバイナリファイルを、実行マシンであるパーソナルコンピュータやハードウェアデバッグ機のマイクロプロセッサに転送することができる。実行マシンに特有のテストは、ここで行われる。このシステムはこのように、ホスト処理するクロスシステムとすることで汎用コンピュータの高速処理、大容量メモリ/ディスク、豊富なソフトウェアツールをフルに利用して、マイクロプロセッサのソフトウェア開発における生産性向上、信頼性向上をねらっているところに特長がある(図2.)。

以下、このシステムの主要なツールを選んで、その機能、処理方式などについて紹介する。

3. COSMO C コンパイラ及び8086 C クロスコンパイラ

3.1 COSMO C コンパイラ

COSMO C コンパイラは、C言語で書かれたソースプログラムをコンパイルし、《MELCOM-COSMO 900II》上で動作する実行モジュールを作成する。このコンパイラでは、C言語の文法仕様、ライブラリ仕様は、

UNIX 互換の標準仕様を採用している。

このコンパイラは、コンパイル時オプションにより、デバッグ用のオブジェクトやカバレッジアナライザ用のオブジェクトを出力することができる。一方、コンパイラの作成したこれらのモジュールをテストするに、テスト支援ツールにより、テストドライバやテストスタグを生成しておく。次にコンパイル済みの被テストモジュールとこれらのドライバやスタグ、及びライブラリをリンクして、テストのために実行モジュールを作成する。実行モジュールのテストは、《MELCOM-COSMO 900 II》のプログラミングワークステーション端末から会話形に行われる。

モジュールテストで検出されたバグの修正は、C言語のソースプログラムに対して行い、常に実行モジュールと対応するように管理する。この過程を繰返し行うことにより、プログラムの完成度を上げていくようにする。なお、このコンパイラは、《MELCOM-COSMO 900 II》用に従来から使用してきたCコンパイラを若干強化して整備した。

3.2 8086 C クロスコンパイラ

8086Cクロスコンパイラは、《MELCOM-COSMO 900 II》の上で、C言語で書かれたソースプログラムをコンパイルし、マイクロプロセッサ上で動作する実行モジュールを生成する。C言語の文法仕様、ライブラリ仕様は、COSMO Cコンパイラのものとは一致している。このクロスコンパイラは、コンパイル時オプションにより、80186、80286 命令や浮動小数点演算用プロセッサ 8087 も利用可能である。

このクロスコンパイラは、DEC 社の VAX/VMS の下で広く利用されているクロスコンパイラ、アセンブラ、リンカを、《MELCOM-COSMO 900 II》上に移植し、更に使い勝手を考慮して、幾つかの機能拡張を行ったものである。クロスコンパイラの大部分は、C言語で記述されているので、VAX/VMS 上のソースプログラムを、《MELCOM-COSMO 900 II》へ転送し、既存の COSMO Cコンパイラでセルフコンパイルした。移植の手順は、図 3. に示すとおりである。

移植した元のクロスアセンブラには、アセンブリリスト出力機能がないので、移植後、クロスコンパイラ、及びクロスアセンブラの両方を改造して機能を追加した(図 4. 参照)。また、クロスリンカの出力するリンクマップも、アドレスによるソートと名前によるソートの両方を行い、見やすく表示するようにした。

クロスコンパイルからマイクロプロセッサ用の実行モジュールを得るまでの手順は、図 5. に示すとおりである。このうち、クロスリンカは、XLINK 86 により、リレータブルな実行モジュールを作り、更に XTO 86 を

0004 C746F80000		*****7 counter = 0;
		mov .w [bp][-8],0
0009	L1:	*****8 while('getchar()' != EOF)
0009 FF36		/line 8
000D E8		push .w _paldin
0010 59		call _getc
0011 3DFFFF		pop cx
0014 7405		cmp ax,-1
		je L3
0018 FF46F8		*****9 ++counter;
0019 EBEB		inc .w [bp][-8]
001B	L3:	jmp.s L1
		/line 9

図 4. 8086 C クロスアセンブリリスト

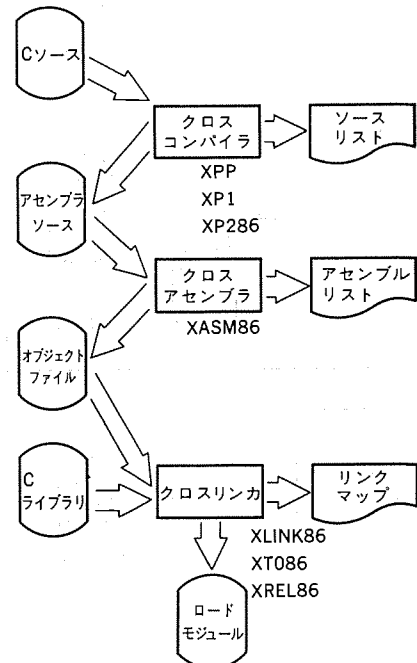


図 5. クロスシステムの手順

通することによって、マイクロプロセッサの実行システムである CP/M、又は MS-DOS 用の実行モジュールなどを生成することができる。

4. テストカバレッジアナライザ

プログラムテストの評価方法の一つに、テスト網羅性をはかる尺度として、テストカバレッジという考え方がある。カバレッジ測定とは、テスト対象プログラム内部の実行制御の全分岐点中の何%の分岐を通過したかにより、テストデータ、テストプログラムの充足度を評価することである。これにより、テストの均一度と冗長度も知ることができる。ここで紹介するカバレッジアナライザは、指標 C1 として知られるカバレッジ計測をC言語のプログラムを対象に実施するツールである。

(1) 基本処理方式

一つのプログラムの内部を制御構造に着目してみると、実行処理部分は分岐命令によって、幾つかの連続した命令の断片(セグメントと呼ぶ)に分割される。これをC言語のプログラムについて考えてみると、そのオブジェクトであるアセンブラソースは BE, BG, B 命令などの分岐命令を区切りとしたセグメントの集合ととらえることができる。図 6. の場合、プログラムは BLE, B 命令を分岐点として三つのセグメント a, b, c に分割されている。このようなセグメントの何%をテストにより実行できたかを評価する。カバレッジ測定のために、その各々の分岐命令の箇所にカバレッジライブラリ呼出しを起こすコードを、

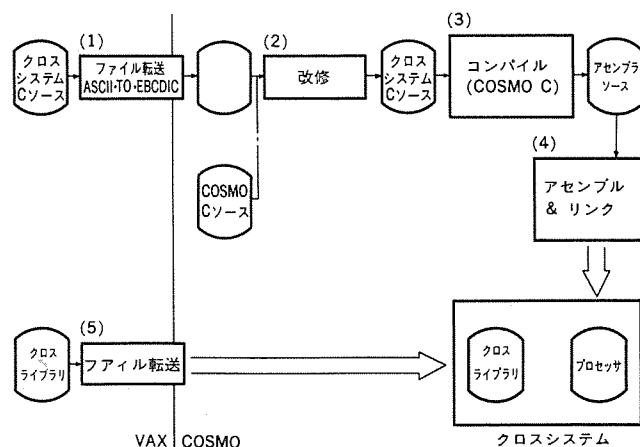


図 3. クロスコンパイラの移植手順

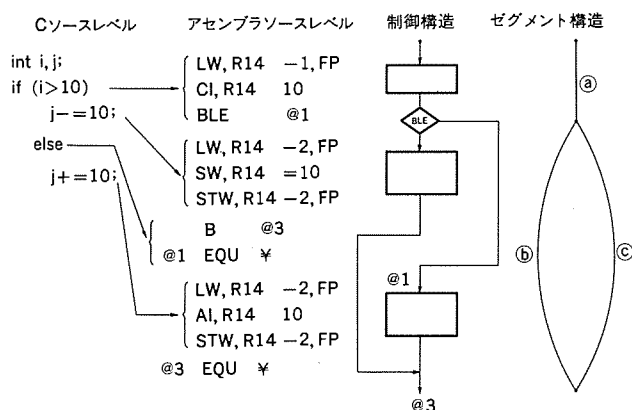


図 6. プログラムの制御構造の例

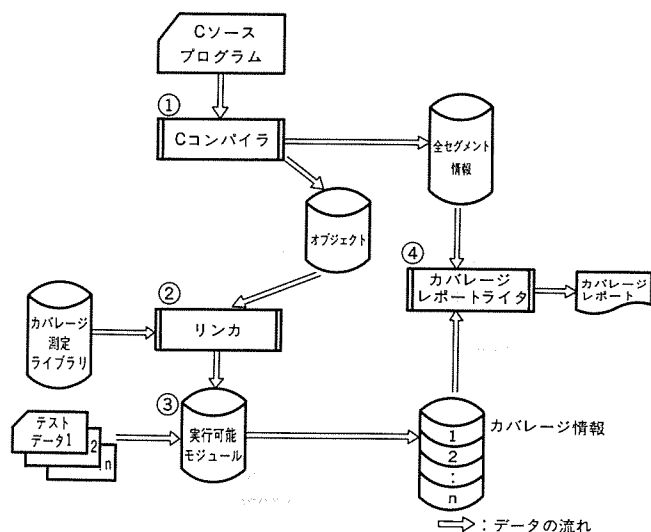


図 7. カバレッジ測定処理の流れ

コンパイル時に挿入する方針をとった。

(2) 処理の流れ (図 7. 参照)

①Cコンパイラにより、カバレッジライブラリ呼出しの命令シーケンスを組み入れたオブジェクトを生成する。また、全セグメント情報ファイルを作り出す。

②①で得たオブジェクトとカバレッジ測定ライブラリをリンクする。

③テストデータ1, 2, ..., nの入力により、各セグメントの通過回数をカウントする。結果は外部ファイルに蓄積して行く。

④カバレッジ測定結果を分析し、レポートする。

(3) 実現方式

①実行時のテストカバレッジ情報を収集するために、コンパイル時オプションで専用のライブラリ呼出しオブジェクトシーケンスを、セグメントの入口ごとの出力する。カバレッジライブラリには、引数として分岐条件が真のときの分岐アドレスとセグメント番号を引き渡す。また、全セグメント情報ファイルには、セグメント番号(0からの通番)とそのセグメントのソース上の位置を表現したセグメント識別番号(Cソース行番号とその行内での分岐番号の組合せからなる)の対応関係を生成する。

②カバレッジライブラリ

オブジェクトから引き渡されてきたセグメント番号に従い、それに対応したカウンタを更新する。また各々のセグメントの実行回数を数え上げて、その結果を外部ファイルに出力し保存する。

③カバレッジレポートライタ

カバレッジの測定結果をレポートするツールである。レポート内容は(a)

1回のテスト終了時に、その回のテストカバレッジ値と過去に実施されたテストのカバレッジの累積値の表、(b)これまでのテストで一度も実行されたことのないセグメントのセグメント番号とセグメント識別番号をリストアップした表、及び(c)今までのテストでのカバレッジ履歴表である。このとき、(b)では指摘されたセグメントがC言語のソースプログラム上、どの部分に対応するかをセグメント識別番号から容易にわかるように工夫している。

5. シンボリックデバッガ

シンボリックデバッガは、C言語で記述されたプログラムを、ソースイメージで会話的にデバッグするためのツールである。

(1) シンボリックデバッガの機能

シンボリックデバッガを利用するには、デバッグオプションの下でソースプログラムをコンパイルする。コンパイラは、出力するアセンブラコードに、行番号やシンボルテーブルなど、デバッグ時に必要な情報を追加する。

デバッグコマンドは、表 1. に示すように、データ調査コマンド、ソース調査コマンド、実行制御コマンドの三つのカテゴリからなり、都合 20 種準備している。デバッグコマンドには、関数の入口と出口で引数の値を示す関数トレースコマンドや、変数の値やアドレスの表示、値の代入を行うコマンドがある。また、現在実行中の行や、コマンドで指定した行に対応するソースステートメントを表示するコマンドもある。実行制御コマンドでは、実行の中断、再開、ステップごとと実行ができる。また、ブレークキーをたたいて、強制中断することもできる。

デバッガは、ライブラリとして作成されており、リンク時に、被デバッグ

表 1. デバッグコマンド一覧(抜粋)

分類	コマンド形式	機能
データ調査コマンド	T	現在位置までの関数の呼出しのネスト関係を引数の値とともに表示する。
	F	制御の流れに従い関数の入口と出口の通過状況を表示する。
	変数(表示指示)	表示指示の形式(16進数, 10進数, 文字など)に従って変数の値を表示する。
	変数=数値	変数に値を代入する。
ソース調査コマンド	P	現在実行中の行のソースステートメントを表示する。
	@P	現在実行中の行の前後 10 行分のソースステートメントを表示する。
	数値P	数値で指定された行のソースステートメントを表示する。
実行制御コマンド	G	現在位置からプログラムを実行する。
	S又は $\overline{N/L}$	ステップごとにプログラムを実行する。
	行番号B	ブレークポイントを設定する。
	BL	ブレークポイントをリストアップする。
	行番号D	ブレークポイントを解除する。
	DA	全ブレークポイントの解除。
	Q	デバッガを終了する。

注 $\overline{N/L}$ は new line

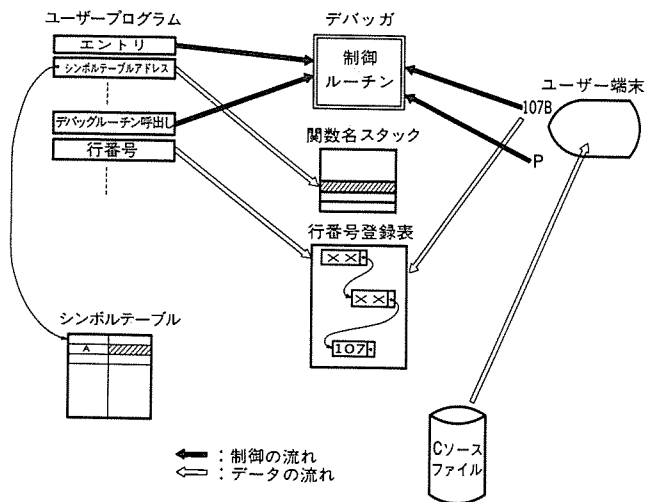


図 8. デバッグ処理の流れ

モジュールと結合して、実行モジュールを構成する。実行の制御は、デバッガが受け持つ。

(2) 表現方式

デバッグオプション付きでコンパイルしたプログラムには、デバッグルーチン呼出し命令、行番号情報が、各ソース行対応に埋め込まれる。変数名、入口名などのシンボルテーブルは、各コンパイル単位の終りに出力される。

図 8. に示すように、B コマンドで実行を中断したい行を指定すると、デバッガは、行番号登録表に行番号を書き込む。行番号登録表は、システムのダイナミック領域をチェーンして使用しており、実用上十分なブレイクポイントを設定できる。C コマンドで実行を開始すると、処理の流れが各行ごとにその行番号を引数として、デバッグルーチンに移る。デバッガは登録表を調べ、一致するものがあればプログラムの実行を中断し、コマンド入力モードになる。一致しなければユーザープログラムに戻って実行を続ける。

コマンド入力モードになったところで、P コマンドを入力すると、今実行が中断されている行のソースステートメントが表示される。W コマンドを使えば、その行の前後 10 行分が表示される。行番号とソースコードとの対応は、標準関数 LSEEK で検索することにより実現している。変数名/表示指示の形でコマンド入力すると、デバッガは、シンボルテーブルを走査して、指定された変数のアドレスを導く。このアドレスの内容を表示指示に従って、10 進数、16 進数、文字などで表示

する。また、T コマンドでは、現在までのサブルーチンの呼び呼ばれ関係を、最も祖先の呼出し地点まで逆上って表示してくれる。このために、専用のスタックを設け、関数の入口と出口で、デバッガが情報の更新を行っている。

(3) デバッガの移植性

シンボリックデバッガは、《MELCOM-COSMO 900 II》以外の他システムへの移植を前提として設計してある。コンパイラの出力するデバッグルーチン呼出し形式、コマンド機能、コマンド形成は汎用的な形としている。また、入出力処理、文字列操作など、システム依存部分も C の標準関数を用いており、移植しやすい。

6. その他のツール

その他のツールとして、ソースシンタックスチェッカ(CHIM: チャイムと呼ぶ)、アサーションチェックプリプロセッサ、テスト支援ツールなどがある。

ソースシンタックスチェッカは、C 言語のソースプログラムの強力な診断ツールであり、C コンパイラを補完する目的で作られている。このツールでは、単一モジュールの診断のほか、複数のモジュールにわたったインタフェース上の診断も可能である。アサーションチェックプリプロセッサは、特定の形式で記述されたアサーションチェック用のステートメントを、C 言語のソースステートメントとして展開するものである。このツールを通さないときは、これらのステートメントは、C コンパイラではコメント行として取扱う。

テスト支援ツールは、被テストモジュールの生成やテストドライバ、テストスタブの生成、テスト用実行モジュールの作成などを支援する会話形のツールである。使い方のガイドンスやコマンドヘルプ機能も付加されている。

7. むすび

効果的なソフトウェア開発のためには、プログラムに適切な作業環境を提供することが重要である。このシステムは、ソフトウェア工学の研究成果を地道に取り入れ、マイクロプロセッサのソフトウェア生産の現場で実際に使用するためのツールとして構築した。このシステムの提供により、マイクロプロセッサのソフトウェア開発の生産性向上、信頼性向上が促進されるとともに、ソフトウェア開発の方法までも変りつつある。

今後は、最適化コンパイラ、ツールデータベース、ローカルエリアネットワークなどの開発ツールを改良整備し、教育普及活動を積極的に進める予定である。また、文書管理、工程管理、品質管理など開発管理ツールの充実も計画している。

基本ソフトウェア開発におけるC言語テストベッドの利用

大江 信 宏*

1. ま え が き

オペレーティングシステムをはじめとする基本ソフトウェア、周辺ソフトウェア、ユーティリティなど多くのプログラム開発をシステム記述言語“C”で行い、そのテストの段階でC言語テストベッドを利用した。以下では、まず2章、3章でC言語及びC言語テストベッドの機能について簡単に説明し、4章で基本ソフトウェア開発において、実際にC言語テストベッドをどのように利用したかを述べる。5章で、実際に利用した上での長所/短所を述べ、6章で、今後利用する上での注意点を述べる。

2. C言語の概要

近年、UNIX^{*1}の普及とともにC言語が広く使われるようになってきており、C言語の文法書もいくつか市販されている。このC言語は、もともと1972年に米国ベル研究所で、Kernigham, Ritchie氏らによって、DECのPDP-11上で動くUNIXオペレーティングシステム用に設計され、実用化されたものである。UNIXでは、オペレーティングシステム、Cコンパイラ/ライブラリ、アプリケーションプログラムのほとんどがC言語で記述されている。

C言語の言語仕様は、PL/I、PASCALほど高級ではないが、構造化プログラミングができ、それに加えて種類の豊富な演算子、PASCALに近いデータ宣言が可能である。ストリング操作や入出力機能などに対する命令はないが、これらは関数をコールするという形で、Cライブラリで実現している。

3. C言語テストベッド

3.1 概 要

C言語テストベッドは、C言語で書かれたプログラムモジュールを効率良くテストするためのモジュールテスト支援ツールであり、UNIXオペレーティングシステムのもとで使用することができる。プログラムのテストにおいて、これまでモジュールの単体テストを十分に行わないまま、次のステップへと移っていくケースがあった。この理由の一つは、モジュールテストを簡単に行える有効なツールが用意されていなかったことがあると思われる。C言語テストベッドは、複雑な構造をもったプログラムの中の一つの対象モジュールをとり出してテストを行えるようにしたツールであり、図1のような構造をもっている。対象モジュール以外の関連モジュールは、あらかじめスタブ^{*2}化しておくことが必要である。この

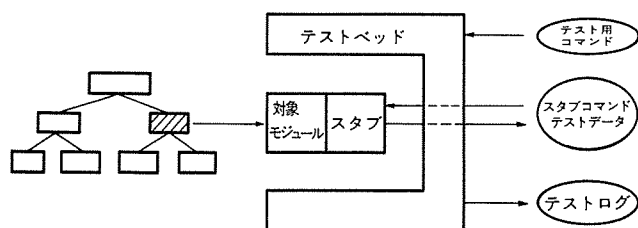


図 1. テストベッドの構造

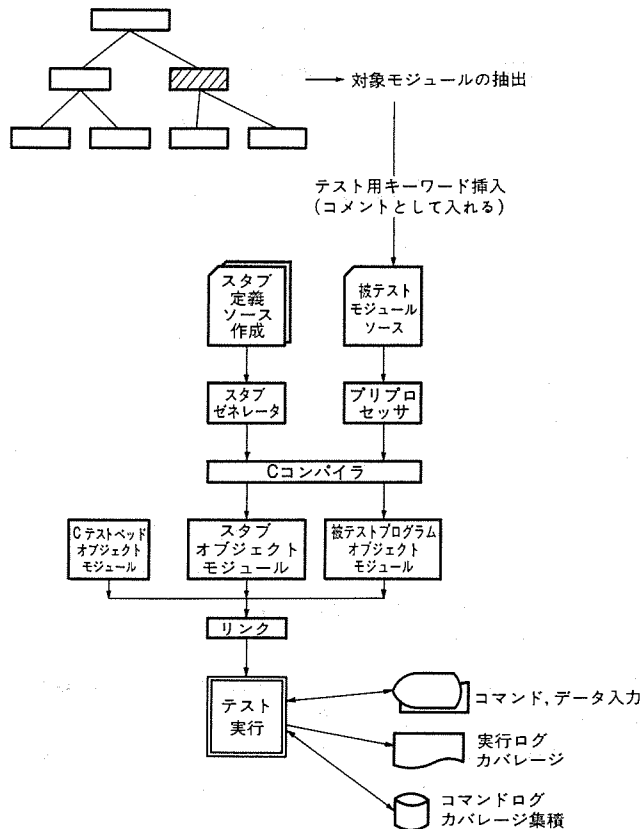


図 2. C言語テストベッド利用時の手順

ためパラメータを与えることによって、スタブを生成するツールが用意されている。

テストの実行は、対話形で進められ、端末から入力したコマンドによって実行の制御が行われる。対象モジュール実行中にスタブ（対象モジュールが呼び出す下位モジュール）が起動されると、特殊なスタブコマンドによる対話が始まる。スタブコマンドはサブルーチンが呼び出されたときの入力パラメータ値のチェックや、再び対象モジュールに戻る際の出力パラメータ値のセットを可能にしている。このテスト結果は、すべてテストログのファイルに保存され、このファイルを使えばモジュールの再テストを一括して行うこともできる。C言語テストベッドは、このほかに次のような機能を備えている。

(1) アサーション

実行時の変数の値をチェックし、その結果を表示

(2) トレース

IF文など特定ステートメントのトレース、変数値のトレース

(3) テストカバレッジ

網羅性を判定する手段として、モジュール内の分岐のうち、何パーセン

*1 UNIXはベル研究所の登録商標である。

*2 テスト用のダミールーチンのことを言う。

トテストしたかの測定及びそのレポート

3.2 C言語テストベッドの利用方法

C言語テストベッド利用時の手順を図2.に示す。この手順は、実際には幾つかのコマンドを実行することでできるようになっている。コマンドは、その中で幾つかのコマンドを実行するコマンドファイルになっており、利用者がたくさんのコマンドを知らなくてよく、また利用環境に応じて変更することも容易である。

4. 基本ソフトウェア開発における C言語テストベッドの利用

4.1 適用範囲と利用形態

C言語で新規に開発したプログラムの多くに対してC言語テストベッドを利用した。この度の基本ソフトウェア開発に際してC言語テストベッドを利用した対象プログラムの種別は、表1.のとおりである。利用形態は、C言語テストベッドですべてモジュールテストを行ったものと、他のマシンでテスト後、テストカバレッジ測定にだけ利用したものがある。後者の比率は、全体の約1割であった。

表 1. C言語テストベッドを利用した対象プログラムの種別

種 類	プ ロ グ ラ ム
基 本 ソ フ ト ウ ェ ア	デ ー タ 管 理 通 信 管 理 シ ャ ー ジ オ ン ト
周 辺 ソ フ ト ウ ェ ア	ジ ャ ン プ 管 理 エ ミ ュ レ ー タ
コ ー テ ィ リ テ ィ	エ デ ィ タ ポ リ ュ ー ム 管 理 フ ァ イ ル 管 理
そ の 他	ラ イ ブ ラ リ

4.2 利用環境

(1) ディスク環境設定

利用者ごとにログイン名を設定し、図3.に示すようなディスク環境設定用のコマンドファイルを作成し提供した。

(2) 運用方法

マシン時間を1日12時間とし、利用者を半分に分けて6時間ずつ割り当てた。使用後は、自分のディスク環境を磁気テープにセーブし、翌日使用時にリストアするようにした。こうした理由は、C言語テストベッドを使うと、スタブ、ログファイルなどでディスクをかなり使用するため、ディスク領域が足りなくなったためである。

4.3 利用方法

C言語テストベッドの利用は、プログラムのモジュールテストの段階であるが、コーディング、モジュールテスト、プログラムテストという開発ステップの中で、C言語テストベッドを利用するための運用規則を設定し実施した。

(1) コーディング

C言語テストベッドを利用するために、キーワードの挿入や、スナップダンピングの指定などをあらかじめ入れておく必要があり、またコーディングスタイルを統一するためにも、コーディング規約を定め、それを守らせた。

(2) 使用手引書の作成

コマンドファイルを整備し、使用手引書を作成して利用者全員がスムーズにテストに入れるようにした。

(3) テストカバレッジの評価

一通りのテストの終わったプログラムの各モジュールのカバレッジを測定し、次のステップのテスト(インテグレーションテスト)に入るための条件を、カバレッジ80%以上と設定し実施した。

5. C言語テストベッド利用上の長所／短所

実際に使用した結果、次のような長所、短所が判明した。

5.1 長 所

(1) テスト用ルーチンをスタブ生成で簡単に作成できるため効率が良

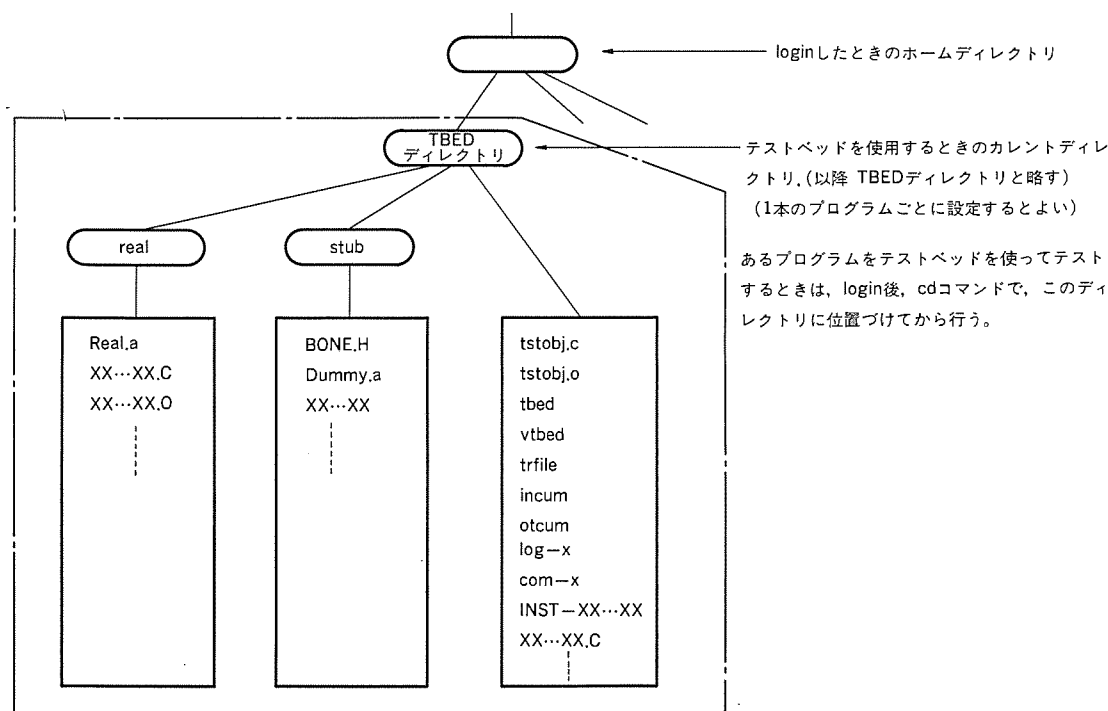


図 3. 基本ソフトウェア開発におけるテストベッド利用時のディレクトリ構造

い。

(2) テスト方法がみんな同じなので、テスト方法の差による品質のばらつきが少ない。

(3) 機能面のテスト(ブラックボックステスト)とカバレッジ利用による論理テスト(ホワイトボックステスト)の併用により、モジュール完成度が高まる。

(4) 対話形のテストの後、入力コマンドログファイルの再利用によりテスト効率が高まる。

5.2 短 所

(1) モジュール数が多くなると、スタブが多くなり手間がかかる。小さいモジュールもすべてテストベッドを利用することはかえって効率を落とすケースもある。

(2) C言語テストベッドのためのコーディング規約があり、既存プログラムの流用では、利用できないケースがある。

(3) 対象テストモジュールの中でエラーになりアボートしてしまったような場合の追求は、Cソースデバッガなどを使用しないとできない。

6. 今後利用する上での注意点

基本ソフトウェア開発におけるC言語テストベッド使用経験から、今後利

用する上での改善すべき点や注意点について述べてみたい。

(1) 標準ライブラリなどでテストマシンに存在しないもの、又は機能が異なるものは、利用者がいちいちスタブ生成してはは無駄であり、標準スタブとして用意しておく方がよい。

(2) カバレッジの測定は、テストの最終段階で管理者が検収として行うのがよい。何故なら、カバレッジのパーセントを向上させるためだけのテスト条件を設定してしまう恐れがあるからである。

(3) モジュール組合せによるテストも、C言語テストベッドを使用して実施すべきである。テストベッド利用によるテスト結果の良否は、画面やログで目で確かめるため、モジュール間インタフェースの誤りを見落とすことがある。

7. む す び

基本ソフトウェア開発にあたり、C言語テストベッドを利用したわけであるが、テスト効率やプログラムの信頼性を高めることができたと確信する。

今後、更にこのC言語テストベッドに限らず、いろいろなソフトウェア開発支援ツールを積極的にとり入れて改善していくことが、増大し多様化するソフトウェアを開発していく上で必ず(須)と考えられる。

リアルタイムUNIX“OS60/UMX”

小川 義高*・臼井 澄夫*・高畑 泰志*・鈴木 裕**・高橋 良岳**

1. ま え が き

計算機の適用分野の拡大と近年のソフトウェア生産量の急激な増大のため、ソフトウェアの開発期間の短縮が要求されるとともに開発コストの低減が強く求められている。ソフトウェア開発の効率化と生産性向上のための計算機システムからのサポートの一つとして、操作性のよい会話形処理が必要とされている。このため、オペレーティングシステムとしては、単に高い実時間処理の性能を保証するだけでなく、端末からの使用者に対しても優れたソフトウェア環境を提供できるものが要求されるようになってきた。

これにこたえるため、このたびオペレーティングシステム OS 60/UMX を新たに開発した。これは、ソフトウェア開発用あるいはワークステーション用 OS として高い評価を得ている UNIX^{*1} を、《MELCOM 350-60》計算機シリーズの実時間処理用オペレーティングシステム OS 60 に組み込んで一体化したものであり、高性能な実時間処理を行うと同時に、端末においては UNIX の環境をそのまま提供できる。以下にその概要を紹介する。なお以下ではオペレーティングシステム OS 60/UMX は UMX と略称する。

2. 開発の背景

2.1 UNIX と実時間 OS

現在、端末作業環境の提供を目的とした OS として最も優れたものの一つに UNIX がある。UNIX は、次のような特長をもっている。

(1) 操作性の良さ

会話形処理における人間の作業性を最も重視した設計となっており、使い勝手がよい。

(2) 豊富な機能

コマンドと呼ばれるユーティリティプログラムが 200 個以上あり豊富な機能をもっている。特にソフトウェア開発支援の機能が充実している。更に、ユーザーがこれらの機能を自由に組み合わせて新しい機能を作り出すことも、非常に容易になっている。

(3) 移植性

大部分が言語“C”で書かれているとともに、ハードウェアに依存する部分も一部に限定されているため、他の計算機で動作させることが比較的容易である。

以上のような特質から、UNIX は多くの計算機で共通に動作する標準 OS としての性格を強めており、今後 UNIX を想定した移植性に優れた流通ソフトウェアも増加してくるものと予想される。しかし、UNIX は多くの長所の反面で、実時間処理については基本構造的に考慮されておらず、このための機能は欠けているといっている。また、処理性能も実時間応用には耐えられないものと考えられる。一般的な使用においても、コンパイラ性能など改善すべき点も多い。

またカーネルを C 言語で記述してあり、移植性が良いということはハードウェアのもつ特長を必ずしも生かしきれないこととなる。

一方、実時間処理用の OS は、イベントドライブ機構、高速プログラム起動、主記憶、常駐のデータやプログラムなど、UNIX のような会話形処理を指向したシステムにはみられない各種の専用機構をもつとともに、高速処理性や信頼性に対しても十分な配慮がなされている。

2.2 新しい OS へのアプローチ

このような点から今後の OS として、ユーザー端末では UNIX が動作し、実時間処理においては実時間用の OS が走るという形態が望ましいものとして考えられる。

この形態には次のような利点がある。

- (1) それぞれの OS が得意な処理を分担することになり、機能的に無理のない構造となる。
- (2) 端末からの会話環境が標準的なものとなるので、操作の習得やソフトウェアの移植が容易である。
- (3) 二つの OS を構造的に独立させておくことができるので、一方のみをバージョンアップすることが容易である。

このような OS の実現方法としては、

- (a) 基本機能のみをもつ核 OS を作り、この上で二つの OS を動作させる方式
- (b) どちらか一方を基本 OS とし、これに他方の OS を組み込む方式

の二つが考えられる。一般には実時間 OS が優れた基本性能をもっていることと、オーバーヘッドを小さく保つために、これを基本 OS として UNIX を組み込む形にするのが全体として最も優れていると考えられる。UMX はこの方式をとっている。

3. 開発のねらい

UMX は次のようなねらいで開発された。

- (1) 高性能な実時間 OS を基本 OS として使用することにより、UNIX システムでは不可能であった高速な実時間処理を可能とする。
- (2) ユーザー端末には生産性の高い UNIX 環境を提供し、ここで会話形処理による実時間制御用アプリケーションプログラムの開発を可能とする。
- (3) 高性能な実時間 OS を基本 OS として用いることにより、UNIX 環境の機能、性能の向上を図る。
- (4) できるだけ実時間 OS と UNIX の相互の独立性を保ち、実時間処理性能を確保するとともに各々のバージョンアップを容易にする。

4. 主な特長

4.1 高速な実時間処理

プロセス管理、主記憶管理、資源管理などの中核処理部に Hash 手法などの高速処理アルゴリズムを取り入れた、実時間処理用のスーパーバイザにより、実時間使用における厳しい要求に十分こたえる機能・性能

*1 UNIX はベル研究所の登録商標である。

を達成している。この実時間処理機能は従来から、OS 60 がもっていたものと完全に同等である。

4.2 端末における UNIX 環境

ユーザー 端末においては、従来の OS 60 と同様に実時間処理系を直接に操作する セッション と、UNIX 環境の セッション のどちらでも ログイン時に任意に選択できる。

UNIX セッション では、UNIX の コマンドインタプリタである shell のもとで UNIX の多彩な コマンドやパイプ機構、コマンドファイルなどを自由に使用することができる。端末から使用できる コマンドの一覧を図 1. に示す。また UNIX 環境の下での実時間制御用 アプリケーションプログラムの作成手順を図 2. に示す。

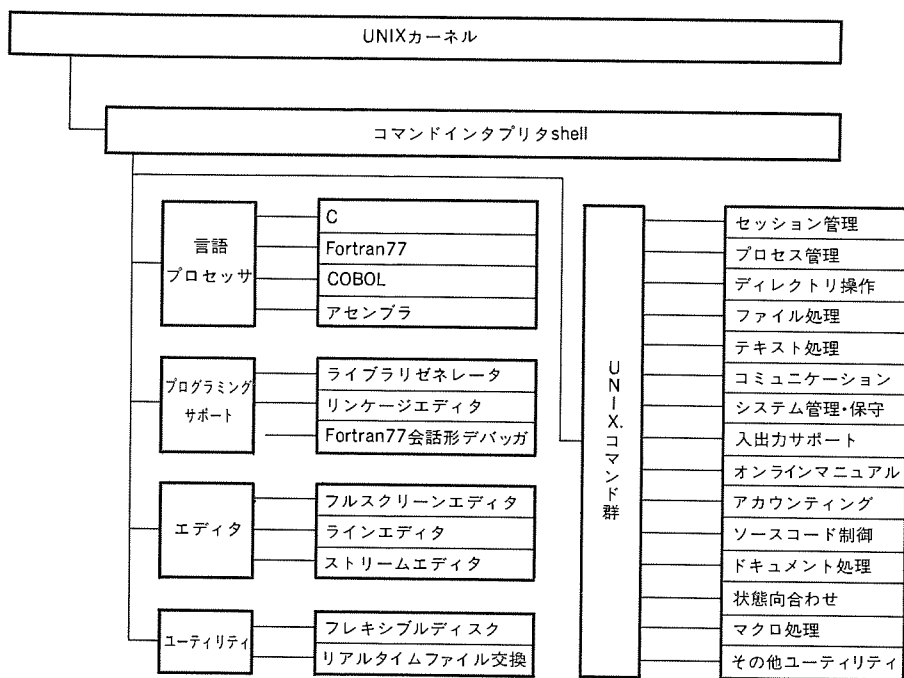


図 1. OS 60/UMX のコマンド

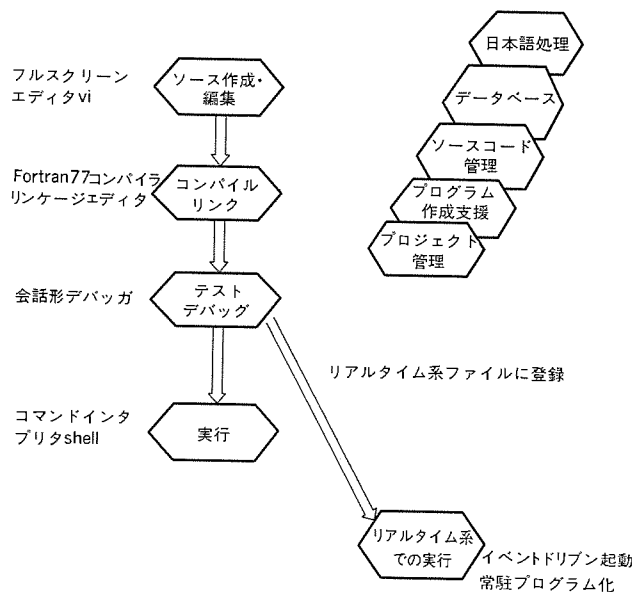


図 2. UNIX 環境の下での実時間制御用 アプリケーションプログラムの作成手順

4.3 UNIX に対する機能強化

UMX の UNIX 環境のセッションにおいては、通常の UNIX のソフトウェア環境をそのまま提供するだけにとどまらず、多くの機能強化や性能の向上を行っている。主なものを以下に述べる。

(1) Fortran 77 コンパイラ 及び 会話形 デバッグ

UMX の Fortran コンパイラは、新たに開発した ANSI 77 準拠のコンパイラであり、論理演算機能や入出力文の拡張などの実時間用機能をもっている。オブジェクトコードはレジスタ配分、DO ループその他に対して最適化が行われ、高速でコンパクトなものとなっている。また、組み込み関数はすべてハードウェア命令として実行し、実行速度の一層の向上を図っている。専用の会話形デバッグは、使いやすいデバッグ機能を提供している。

(2) プログラムローディングの高速化

UNIX のプログラム起動は、ばらばらに分散配置されたファイルから行われるため、オーバーヘッドが大きく遅い。このため、OS 60 の機能を利用してシステムプログラムは OS 60 の区分編成ファイルにおき、プログラムローディングの高速化を実現した。この方式によるローディング時間の短縮を図 3. に示す。

(3) OS 60 の コマンドの使用

コマンドインタプリタ shell から OS 60 の各種のコマンドも、そのまま使用することができる。これにより、実時間処理系のファイルやプログラムを UNIX セッション内で直接に操作することができる。

(4) フルスクリーンエディタ

従来のベル研版 UNIX (SYSTEM III) ではサポートされていなかったフルスクリーンエディタ“vi”も使用できる。vi は、カーソルの自由な移動、ほとんどワンタッチの編集機能など優れた操作性をもち、ラインエディ

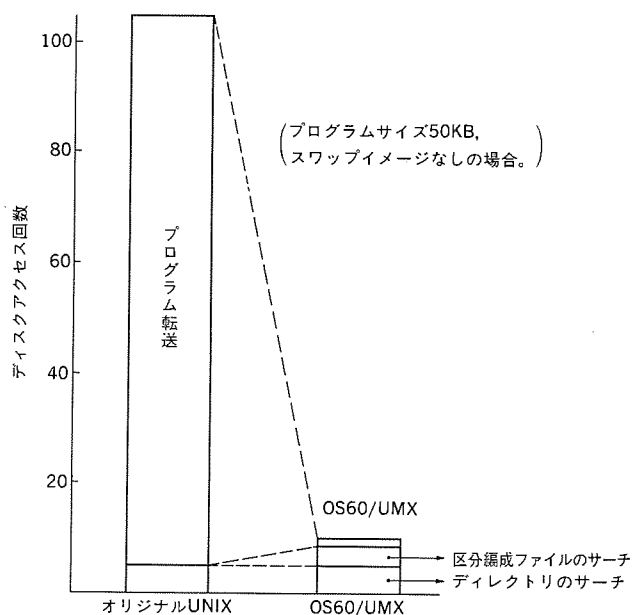


図 3. プログラムロードの高速化

タに比べ、大変使いやすいものとなっている。

(5) ファイルアクセスの高速化

ファイルブロックサイズの変更と、ディスクドライバにおけるアクセス手段の改良により、ファイルアクセスの高速化を図った。また端末アクセスも改良してオーバーヘッドを低減した。

(6) リレーショナル形データベース

理解しやすいリレーショナル形データモデルと高水準のユーザーインタフェースを提供する。日本語の取扱も可能である。

5. 実現方式

5.1 構造

UMXの全体構造を図4.に示す。ハードウェアの操作やシステム全体の管理など基本的な処理は、実時間OSのスーパーバイザであるOS60カーネルが一元的に行う。ここに、UNIXのカーネルが組み込まれ、ユーザープログラムに対してシステムコールを提供する。このシステムコールはUNIXと完全に互換性を保っているため、コマンドインタプリタ shell や各種のUNIXコマンドがそのまま動作することができる。

5.2 UNIXの移植

UNIXの移植は、大略して次のような手順で行った。

(1) カーネルの改修

カーネルは、入出力用のドライバを含め約2万行のソースからなり、うち9割はC言語で書かれている。このうち、計算機のハードウェア(CPU、入出力装置など)を直接に操作あるいは利用している部分を改修した。更にカーネルの中核的な処理は実時間OSのスーパーバイザに移し、全体の高速化を図った。

(2) ファイルシステムの生成

OS60のファイル管理単位である“ボリューム”の中に、UNIXのファイルシステムを生成し、この中に必要なディレクトリやコマンドを登録した。これらの作業はUNIXのファイル管理ツールを改造しOS60のユーティリティとして動作させて行った。図5.にUNIXファイルシステムの配置を示す。また図6.にその生成

方式を示す。

(3) コマンドの移植

UNIXのコマンドは、C言語で記述されており、システムコールとランタイムライブラリのみを使用して動作している。このためランタイムライブラリを用意すれば、ほとんどはそのままコンパイルし直すだけで移植できたが、ごく一部にハードウェアのアーキテクチャに依存するものがあり、

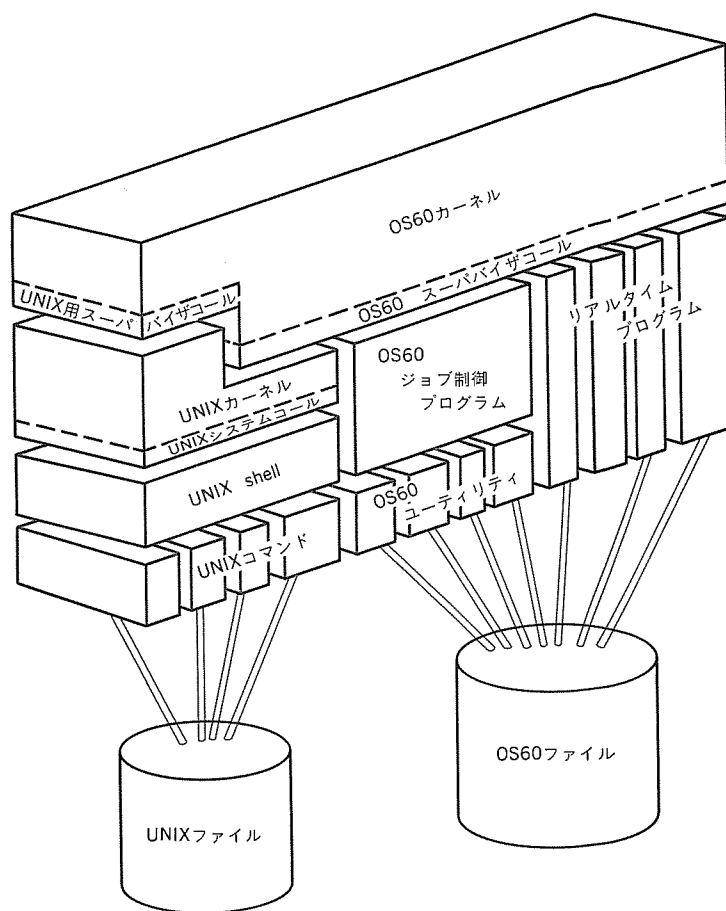


図4. OS60/UMXの全体構造

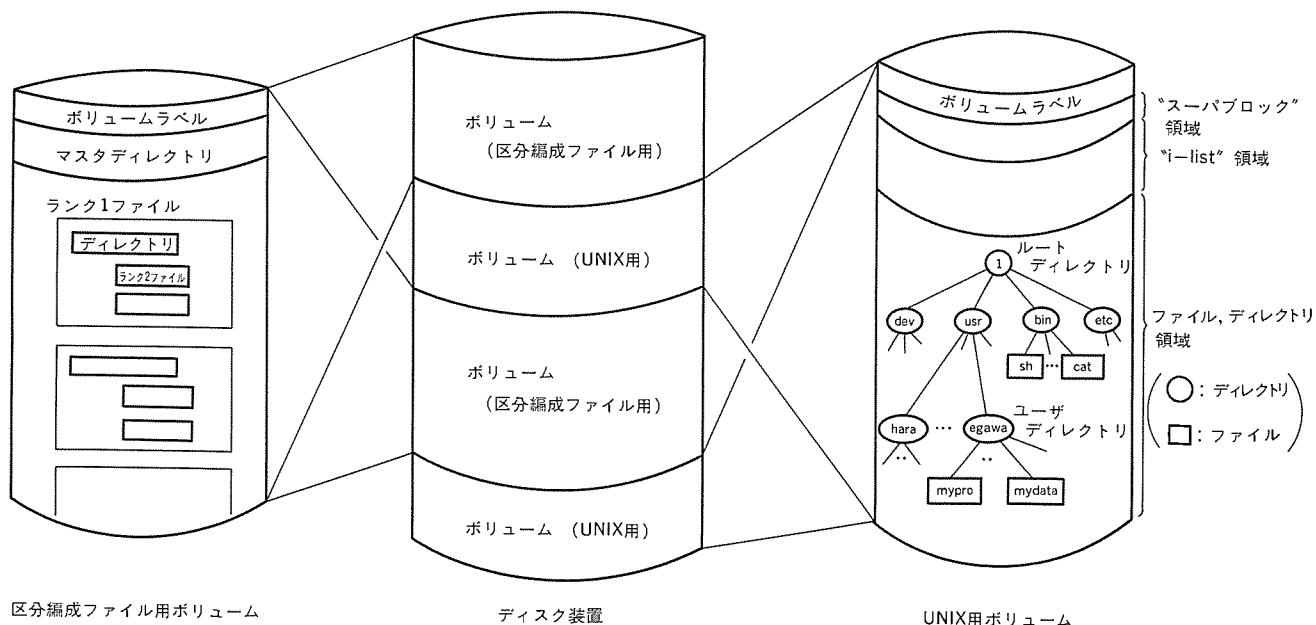


図5. UNIXファイルシステムの配置

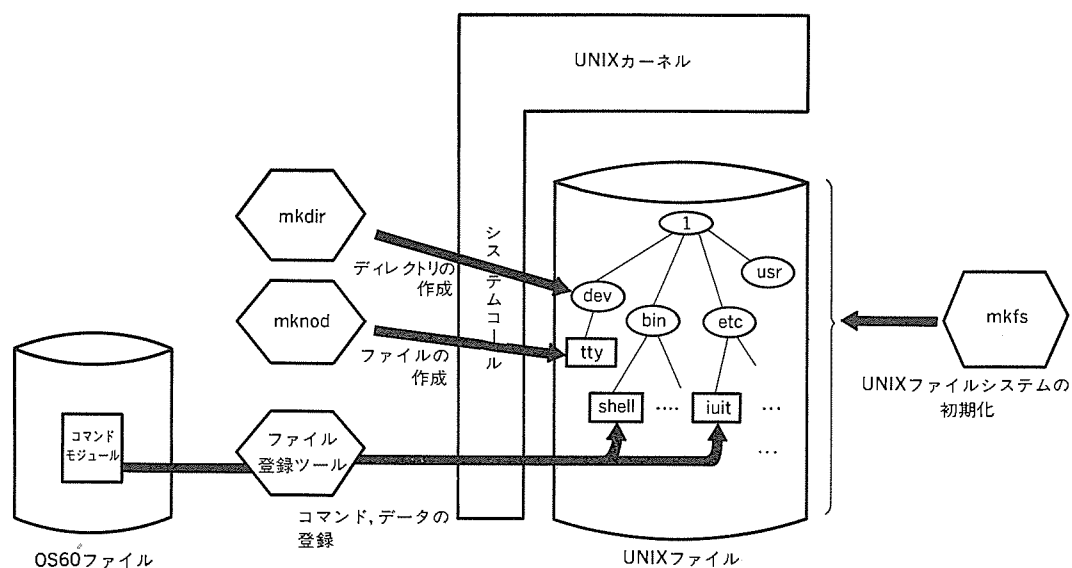


図 6. UNIX ファイルシステムの生成方式

これらについては改修を行った。

の実現に努めていく所存である。

6. む す び

参 考 文 献

高速な実時間処理機能と端末からの優れた会話形処理環境を、同時に可能とした OS である OS 60/UMX について、その特長と方式の概要を紹介した。

OS に対する多様なニーズにこたえることは計算機システム開発における大きな課題の一つであり、今後も更に高度で使いやすい機能

- (1) Bell Laboratory : UNIX User's Manual, Bell Laboratory
- (2) 石田 : UNIX, 共立出版 (昭 58)
- (3) 小川ほか : 工業用計算機《MELCOM 350-60》の基本ソフトウェア, 三菱電機技報, 57, No. 4 (昭 58)

燃料電池発電プラント制御システム

佐々木 明*・水本 洋一**・日比野 学***・大西俊一郎***

1. ま え が き

近年のエネルギー事情から、エネルギー源の多様化及び省エネルギー性の向上が望まれている。一方、大容量電源設備の偏在化、更に送電系統設備の長距離化などの問題が発生しているから、電源設備は需要地に直結した小規模分散形の傾向を強めていくと予想される。

燃料電池発電システムは発電効率が40%以上であり、熱利用も含めた総合熱効率は80%以上が期待できる。また規模でも効率が高く、環境保全性も良好なので都市内設置の分散形発電所に適する。更に部分負荷運転でも高い発電効率を維持し、負荷変動に対する応答が速く起動時間も短いので、ピーク負荷用電源に適した特性も持っている。

現在、天然ガスを原料とするりん酸形燃料電池発電システムの実用化時期が最も早いと考えられており、各方面で精力的な研究開発が進められている。当社では電力事業用のMW級りん酸形燃料電池発電システムに重点をおいて、燃料電池本体、燃料改質装置、直交変換装置及び制御装置から構成される全システムの研究開発を実施している。昭和55年に交流出力500Wの発電プラントの試作・運転を行い⁽¹⁾、その後、要素技術の進展をもとに直接デジタル制御(DDC)方式の制御装置を設置した50kWプラントを昭和56年に開発した⁽²⁾。更に商用プラントを想定して、熱回収系統の追加及び制御装置の機能向上を図った50kW-MKⅡプラントを昭和58年初期に完成した。

実用プラントの制御仕様は次の4項目に集約される。これらの要求を完全に達成し得る新しい制御システムを開発することが基本的目的である。

- (1) プラント全体の発電効率は40%以上
- (2) 負荷変動に対する応答は15秒以下
- (3) 冷機状態からの起動時間は4時間以下
- (4) プラントは無入/遠隔運転可能

MKⅡプラントでは特にプラント全体の自動運転方式を検証し、負荷変動に対する応答性の向上を図ることに努めた。

本稿では、まず制御対象と考えた場合の燃料電池発電プラントの特性を述べる。次に50kW-MKⅡプラントを例として、運転仕様を満たすために要求される制御システムの構成と機能、更に自動運転結果について報告する。

2. 燃料電池発電プラント

2.1 プラント特性

燃料電池本体の電気的な負荷に対する応答は0.1秒以下と極めて速い。この特長に着目して、負荷追従性が高い電源への適用が期待されている。独立電源として使用した場合は更にこの特長が発揮される。燃料電池発電プラントは図1.にプラントの系統例を示すが、化学プラントと発電プラントが結合したものと見ることができる。したがって、温度、圧力及び流量に代表されるプロセス量の応答速度と、電流、

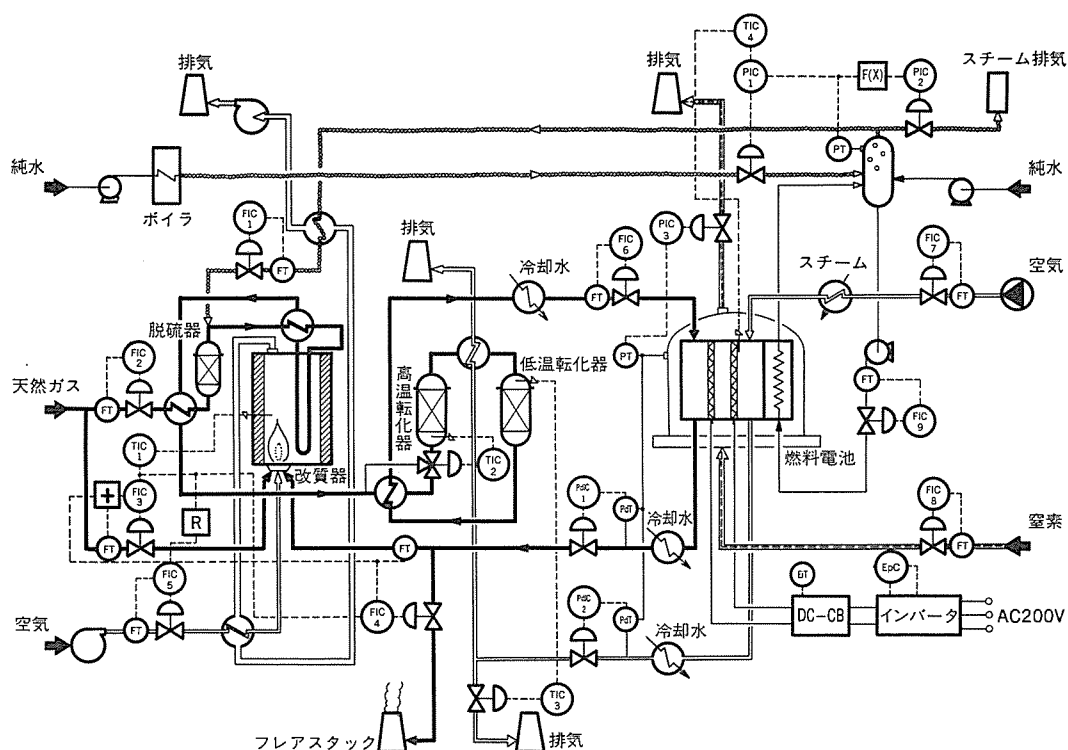


図 1. 50 kW-MKⅡ プラントの系統図

電圧などの電気量の応答速度の差を制御系が整合させなければ、円滑な運転を実現することができない。

さて、主要機器である燃料電池本体の発電特性、すなわち負荷（電流）に対する電圧は、動作圧力、燃料あるいは空気利用率（供給した水素あるいは酸素が消費される割合）そして動作温度に強く依存する。まずこれらの諸特性とプラントとの関係について述べる。

2. 1. 1 動作圧力

動作圧力の高い方が燃料電池本体の発電効率は高いが、高圧運転を実施すると圧縮機などの補機損も増加する。したがってプラント全体の発電効率を最大にする最適な動作圧力が存在する。

一方、燃料電池本体は電解液マトリクス層の耐圧及びマニホールドと燃料電池本体の間のシール耐圧が高くとれないので、燃料電池本体を圧力容器（電池筐体）に入れ、窒素ガスなどを供給することにより雰囲気全体と燃料及び空気の圧力を同一にする構造として、所要の動作圧で運転する。このため、電池きょう（筐）体—燃料極及び電池筐体—空気極の差圧に関する定値性能が、プラントを安全に運転するための重要な要因となる。

2. 1. 2 利用率

燃料利用率及び空気利用率と発電効率は反比例の傾向にある。すなわち利用率を上げると、燃料中の水素分圧あるいは空気中の酸素分圧が低下するため、燃料電池本体の出力電圧が低下する。プラント全体の運転効率と電池特性を勘案して、燃料電池発電プラントを所定の利用率で負荷追従運転するには、燃料流量及び空気流量に対する迅速な追値制御系が要求される。しかし燃料系統には燃料改質装置という大きな遅れ要素がある。すなわち負荷指令に対し、発生する改質ガス流量の追値性能が、プラントの応答を規定する最大の要因である。また、燃料改質装置で進行する水蒸気改質反応の効率は、改質温度に最も左右される。したがって、負荷変動に対する改質器温度の定値性能も重要である。

2. 1. 3 動作温度

動作温度の高い方が燃料電池本体の発電効率が低い。しかし使用材料の耐食性を考慮して、動作温度は190～205℃付近に設定される。負荷変動にかかわらず動作温度を一定にするため、水冷方式においては、電池本体に冷却配管を設置し加圧水を循環させ、冷却水温度（飽和水蒸気温度）を冷却系統の圧力（飽和水蒸気圧力）により調節する。また、冷却系統で発生する水蒸気を燃料改質装置に供給して、プラント全体の熱効率を向上させる。一方、インバータは負荷指令に応じて電力制御を行うため、電池の出力電圧が低下すれば出力電流値を増加させるので、電池本体の発生熱量が増加する。このように負荷変動は最終的に温度変動となって表れるので、動作温度の制御が必要になる。

2. 2 制御量と制御仕様

燃料電池発電プラントに対する制御仕様は、追値性を要求するものと定値性を要求するものに大別される。更に追値性能は負荷変動に対する応答性に関係する。そして定値性能はプロセス量の応答と電気量の応答との間の遅れを補償するものであり、プラント運転の安全性に関係する。またこの制御仕様は、局所的なもの、局所的なものの部分的な組合せ及びプラント全体に関するものに階層化できる。

また、燃料電池発電プラントにおける主な制御量と制御仕様を表1に示す。また、いくつかの制御量に対しては、部分的な制御量の組合せとして、比率制御あるいはカスケード制御を導入した。また、プラント全体の負荷追従運転に関係する主要な制御量は、有効電力、

表 1. リン酸形燃料電池発電プラントにおける代表的な制御量と制御仕様

直接的な制御量	仕様	関連する制御量	仕様	リンク方式	設定例
天然ガス流量	追 値	スチーム・カーボン比	定 値	比 率	3.5
水蒸気流量	追 値				
燃料改質装置圧力	(定 値)	—	—	—	6.0 kg・cm ⁻² G
改質ガス流量	追 値	燃料利用率	定 値	—	75%
空気流量	追 値	空気利用率	定 値	—	60%
燃料電池筐体圧力	定 値	—	—	—	4.0 kg・cm ⁻² G
筐体—燃料極差圧	定 値	—	—	—	100 mm 水柱
筐体—空気極差圧	定 値	—	—	—	50 mm 水柱
改質器温度	定 値	—	—	—	800°C
改質器バーナ燃料流量	追 値	空気過剰率	定 値	カスケード比率	1.1
余剰ガス放出流量	追 値				
改質器バーナ空気流量	追 値				
ターボコンプレッサ助燃量	追 値	全空気利用率・空気過剰率	追 値	—	—
燃料電池温度	定 値	—	—	カスケード	190°C
冷却系統圧力	追 値	—	—		—
冷却水循環流量	(追 値)	—	—		—
冷却水タンク液面	(定 値)	—	—	—	—
有効電力	追 値	直流電流	追 値	—	—
無効電力	追 値				
周波数	定 値	—	—	—	60 Hz

天然ガス流量、水蒸気流量及び空気流量である。表1に示した制御量以外に、各熱交換器の温度がある。

3. 50 kW-MKⅡ 制御システムの構成

3. 1 50 kW-MKⅡ プラントの構成

50 kW-MKⅡ プラントの系統図を図1に示す。燃料改質装置は脱硫器 (Desulphurizer)、改質器 (Reformer)、高温CO転化器 (High Temperature Shift Converter) 及び低温CO転化器 (Low Temperature Shift Converter) で構成される。更に、電池本体の冷却系統から発生する水蒸気を回収して改質反応に利用している。また燃料電池本体の燃料極から排出される余剰ガスを改質器のバーナで燃焼させている。各反応器間には、プロセスガス相互で熱交換を行わせる再熱交換器を設置し、温度の定値性と運転効率を向上させている。なお、燃焼電池本体の空気極から排出される空気をCO転化器間の温度制御に利用している。

3. 2 コントローラの構成

一つの設定量 (Set-point Variable: SV) に対し、対応する一つのプロセス量 (Process Variable: PV) を検出・比較して、対象に応じた効率で操作量 (Manipulated Variable: MV) を出力することにより、設置量とプロセス量の偏差を除去するフィードバック系を1ループ制御系と呼ぶ。MKⅡプラントで使用した1ループ制御系の代表的な構成を図2に示す。操作量は電流信号 (DC4～20 mA) で出力し、プロセス量は上位コントローあるいは計測系との共用を考慮して電圧信号 (DC1～5 V) で入力する。

MKⅡプラントのコントローラは、階層化し分散化する構成とした。制御システムの構成を図3に示す。流量制御 (FIC) に14ループ、圧力制御 (PIC) に3ループ、差圧制御 (PdIC) に2ループ及び温度制御 (TIC) に5ループ、合計24ループの1ループ制御系を設置した。更に、負荷運転時に負荷指令に応じて各々の1ループ制御系に設定量を与え、プラント全体の協調をとるホストコントローラを設置した。またホストコントローラ

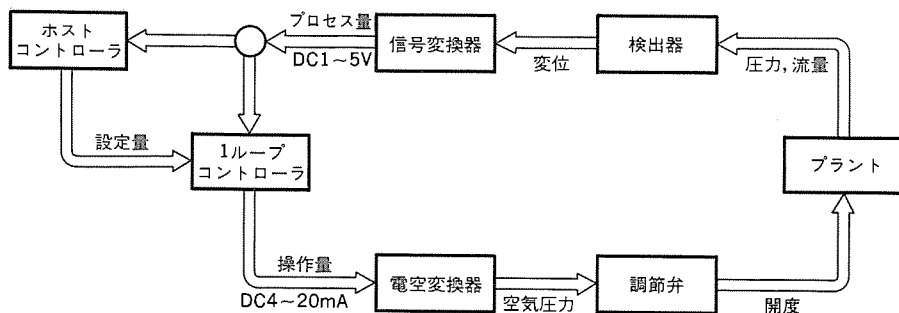


図 2. 1 ループ制御系の構成

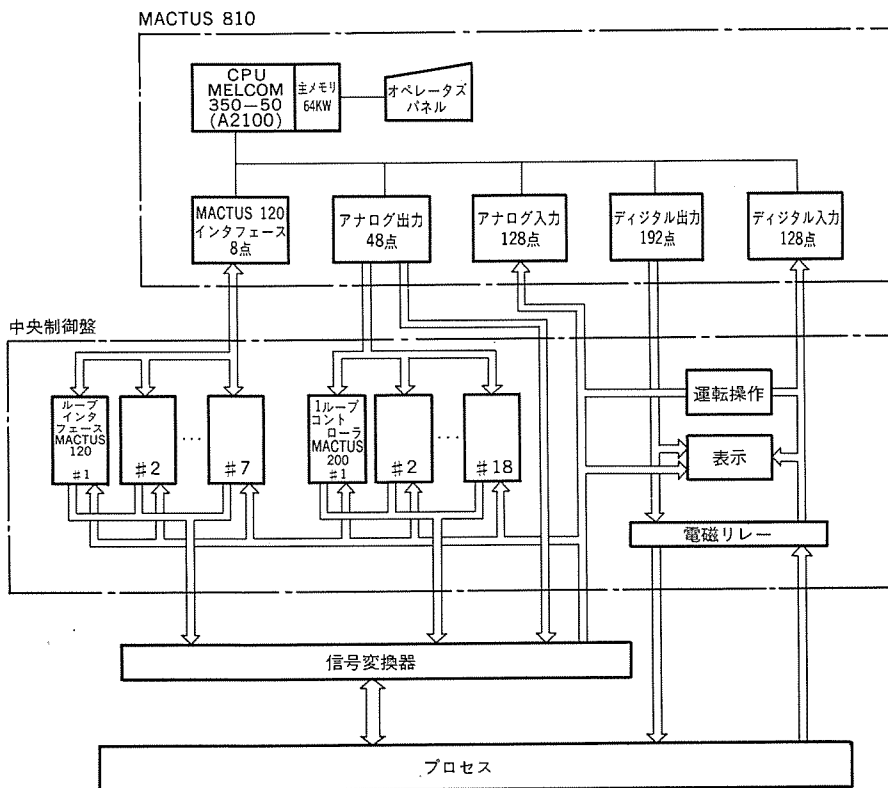


図 3. 制御システム構成図

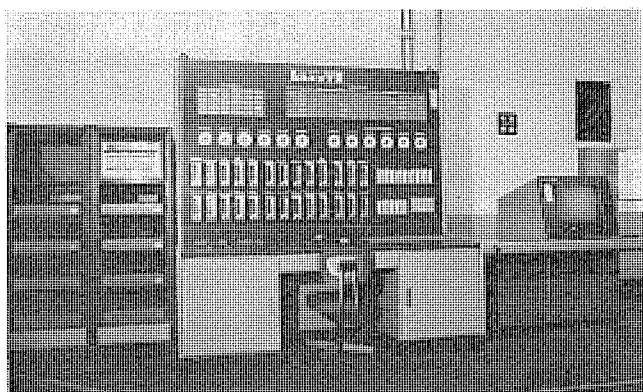


図 4. 中央制御盤全景

には、起動停止などのシーケンス制御も負担させた。

ホストコントローラには《MACTUS 810》を使用し、1ループコントローラには《MACTUS 200》を18台設置し、ホストコントローラとのデータ転送はアナログ信号で行った。なお、プラント全体の負荷追従運転に最も重要な1ループ制御系はホストコントローラに負担させ、システムダウン時のバックアップ用にループインタフェース《MACTUS 120》を7台設置した。

3.3 中央制御盤の構成

MK II プラントの中央制御盤全景を図 4. に示す。中央制御盤の垂直面には、運転状態表示灯、アラーム表示灯、指示計、1ループコントローラ及びループインタフェースを配置した。制御卓上にはスイッチ類を配置し、操作員は運転モードの選択、シーケンス制御関連の操作、手動・調整運転時の各遮断弁の開閉操作、ポンプなどのON-OFF操作及び緊急停止操作を容易に行えるようにした。

また、プラントの保護・監視装置として、カラーグラフィックCRT付きのモニタステーション《MACTUS 670》を設置し、操作員にプラント情報を視覚化して提供できる構成とした。更に《MELPLAC-10》をホストコントローラとは独立に設置し、ホストコントローラの異常時にもプラントを安全に停止できる体制を整えた。

4. 50 kW-MK II 制御システムの機能

4.1 運転の自動化

燃料電池発電プラントの運転には、起動・停止時のガス切替、負荷運転時の流量制御、差圧制御など複雑で微妙な操作を行う必要がある。更に可燃性ガスを処理する化学プラントであるから、危険防止及びプラント保護のため非常停止時に系内を迅速に窒素パージする必要がある。また分散形発電所として適用された場合、給電指令所からの遠隔操作或は無入運転が要求される。これらを想定して、MK II プラントでも、大幅な運転の自動化を図った。

MK II プラントで可能な運転モードを表 2. に示す。半自動運転とは、シーケンス制御中の限定ステップ内だけ自動運転し、次のステップに移行するかどうかは人間が判断するものとした。また MK II プラントでは各構成要素の単体運転にも対応できるように、システム運転以外に燃料改質装置の単体運転（FP 単独）及び燃料電池本体の単体運転（FC 単独）のモードを設けた。

一方、運転方案の確立を目的としているので、制御装置には相当な柔軟性を持たせた。運転方案の変更は、ほとんどソフトウェアのみ

表 2. 運転モード

	システム	FP 単独	FC 単独
起 動	◎○△	◎○△	◎○△
通常停止	◎○△	◎○△	◎○△
非常停止レベル1（軽故障）	○	◎	○
非常停止レベル2（重故障）	○	◎	○
緊急停止（駆動源遮断）	△	△	△
通常負荷運転	◎	—	—

◎：自 動

○：半自動

△：手 動

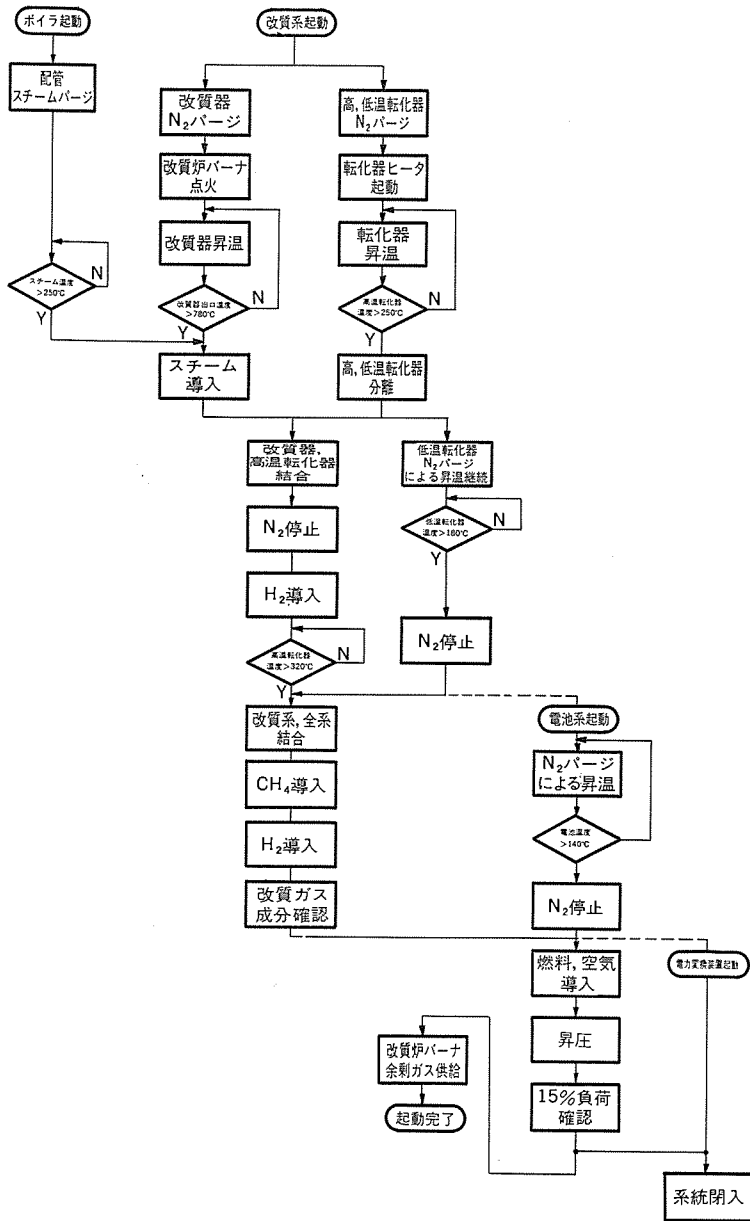


図 5. 50 kW-MK II プラントの起動シーケンスフローチャート

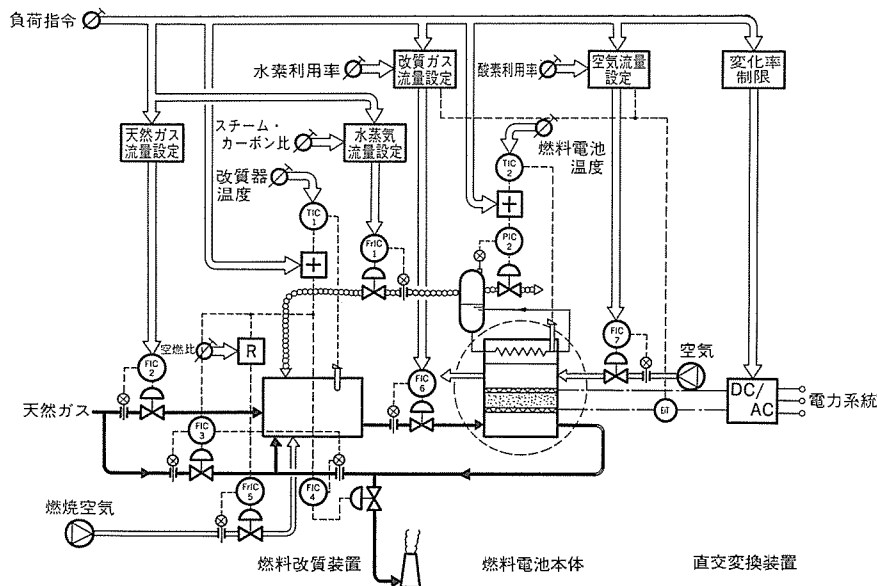


図 6. 50 kW-MK II プラントの制御方式

の変更で処理可能とした。また手動—自動の切換及びシーケンス制御—ループ制御の切換は広範囲に許容し、最新のプラント情報に関するデータベースをもとに、バンプレス (Bumpless) を切換を実現した。

4.2 シーケンス制御

起動・停止などのシーケンス制御で最も重要な点は、ガスの導入順序及び切換に伴う各部の温度設定である。したがって設定温度の高い機器ほど早く起動し、タイミングをとりながらプラント全体を最小負荷運転状態にする必要がある。

MK II プラントにおける冷起動シーケンスを図 5. に示す。この例において、燃料改質装置の昇温時間及び電池筐体の昇圧時間がプラント全体の起動時間を規定する。また、シーケンス制御中も最下位の 1 ループコントローラを PID 動作による自動制御状態とし、設定量を操作することで各ステップを進行させた。

4.3 通常負荷運転

MK II プラントで採用した全系の制御方式を図 6. に示す。負荷追従運転に密接に関係するプロセス制御系は、水蒸気流量 (FIC-1)、天然ガス流量 (FIC-2)、バーナ燃料流量 (FIC-3)、余剰ガス流量 (FIC-4)、バーナ空気流量 (FIC-5)、改質ガス流量 (FIC-6) 及び空気流量 (FIC-7) の合計 7 ループである。これらの 1 ループ制御系はホストコントローラに負担させ、補正演算、ループ間の協調などを容易にした。また改質器温度制御系 (TIC-1) は、バーナへ投入する燃料流量及び空気流量制御系を二次ループとし、燃料電池温度制御系 (TIC-2) は冷却系の圧力制御系 (PIC-2) を二次ループとするカスケード制御系を構成した。なお、これらの温度制御系に対しては負荷指令に応じたフィードフォワード操作量をホストコントローラから直接与え、温度の定値制御性能の向上を図った。

操作員が定数として与えるべき情報は、改質器温度、燃料電池温度、水素利用率、酸素利用率、スチーム・カーボン比及び空燃比である。これらの情報はすべてホストコントローラに入力され、中央制御盤からの負荷指令にもとづき各 1 ループ制御系の設定量を演算した後、上記 7 ループの PID 演算を行いプラント側に出力した。なお、プロセス量の応答と電流量の応答との間の協調をとるため、中央制御盤からのステップ状の負荷指令に対し一次遅れ処理を施した。

4.4 保護・監視
燃料電池発電プラントでは、保護・監視系の機能検討も重要な課題であり、以下に示す項目について重点的に設計を行った。

4.4 保護・監視

燃料電池発電プラントでは、保護・監視系の機能検討も重要な課題であり、以下に示す項目について重点的に設計を行った。

- (1) 圧力スイッチなど保護用検出器の設置
- (2) 駆動源 (計装空気圧力・制御電源) 断のとき、弁開度を安全側にするような弁の選定
- (3) 緊急遮断弁の設置
- (4) 停止モードの分類
- (5) 保護用計算機の独立

保護動作系統図を図 7. に示す。非常停止の内、レベル 1 は軽故障で発生し原則として通常停止と同じである。レベル 2 は重故障で発生し保護用計算機がプラントを停止する。レベル 3 は駆動源の一つのスイッチ

で直接遮断し、機械的に弁を安全側に開、又は閉する。更に停止モードと停止シーケンスを図 8. に、停止モードとプラントの異常要因を表 3. に示す。

5. 運 転 結 果

MK II プラントの運転試験は、2 章で述べた制御仕様に対しプラントを最適状態に設定するとともに、特に発電プラントの起動特性及び負荷追従性について、各種条件において実施した。

運転特性の一例として、起動特性例を図 9. に、負荷応答例を図 10. に示す。図のように、起動時間は 4 時間以内、負荷変動に対するプラントの立上りは 100% 1 分以上を得ている。この運転結果は、前述のプラントに要求される制御仕様をほぼ達成している。今後、制御パラメータの最適化により、更に特性は向上すると考えられる。また、プラントの応答を高めるために必要な構成機器の熱容量及び容積に対する貴重

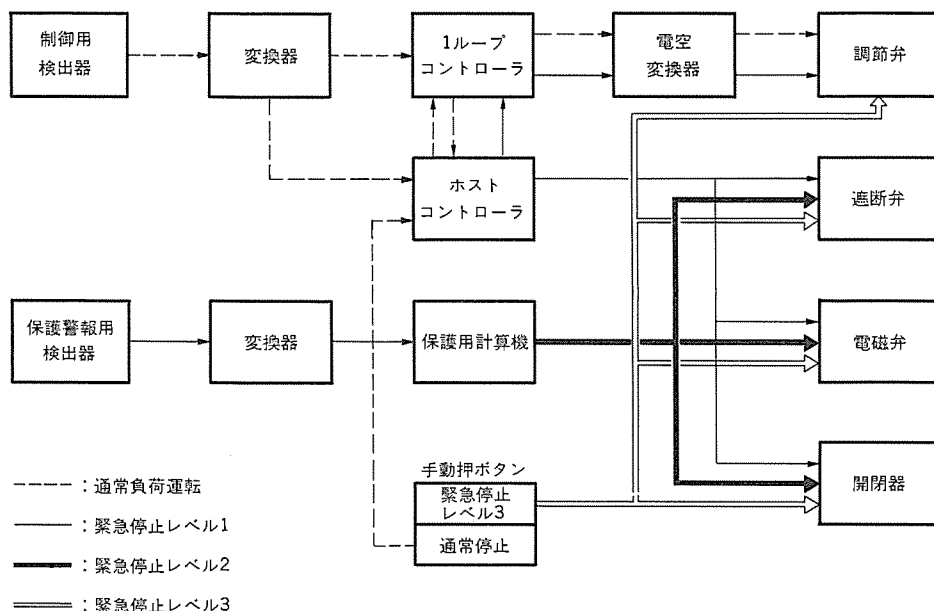
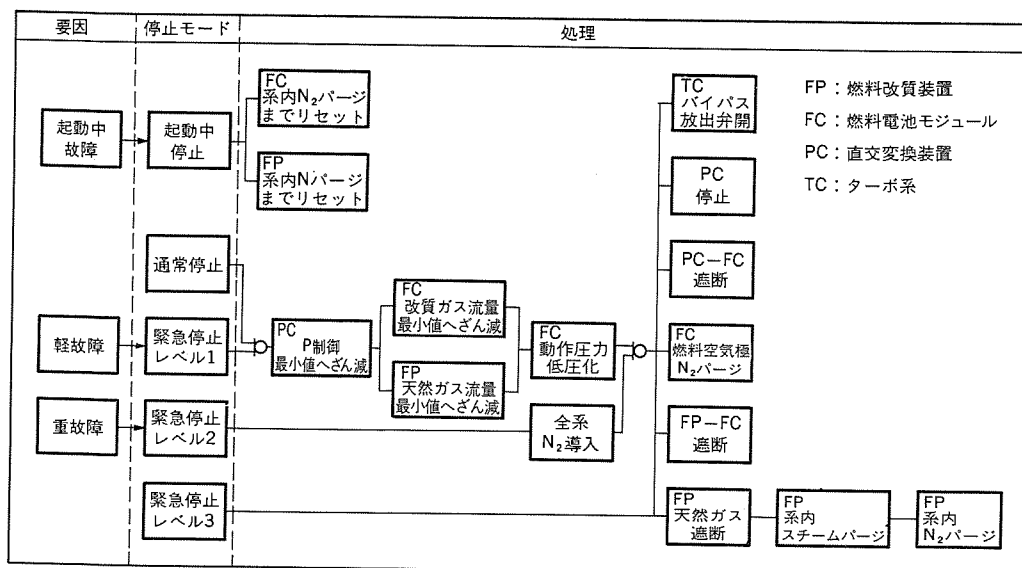


図 7. 保護動作系統図

表 2. 停止モードとプラントの異常要因

緊急停止レベル 1	緊急停止レベル 2	緊急停止レベル 3	起 動 中 停 止
A 燃料改質装置異常 (1) 改質器出口ガス温度高/低 (2) 高温転化器出口ガス温度高/低 (3) 低温転化器出口ガス温度高/低 (4) 脱硫器出口天然ガス温度高/低 B 燃料電池モジュール異常 (1) 燃料電池余剰ガス温度高 (2) 燃料電池出口空気温度高 (3) 燃料電池冷却系温度高 (4) 燃料電池筐体一燃料極差圧高 (5) 燃料電池筐体一空気極差圧高 C 電気系異常 (1) 直流不足電圧 (2) 直流過電流 (3) 交流不足電圧 (4) 交流過電圧 (5) 交流過電流 (6) 交流周波数異常 (7) インバータ内部故障 (8) 補装回路過電流	A 燃料改質装置異常 (1) 改質器バーナ失火 (2) 改質器反応管表面温度過昇 (3) 改質器炉内圧力高 (4) 水蒸気比低下 (5) 反応系圧力過昇 (6) 天然ガス圧力低下 (7) 冷却水圧力低下 (8) 改質器排ガス中可燃性ガス混入 (9) 燃料電池余剰ガス流量過大 B 燃料電池モジュール異常 (1) 燃料電池セル温度過昇 (2) 燃料電池余剰ガス温度過昇 (3) 燃料電池出口空気温度過昇 (4) 燃料極一空気極差圧過昇 (5) 燃料電池筐体内圧過昇 (6) 燃料電池冷却系セパレータ内圧過昇 (7) 燃料電池出口空気中可燃性ガス混入 (8) 燃料電池筐体内可燃性ガス混入 (9) 燃料電池冷却系ポンプ停止 C ターボコンプレッサ異常 (1) ターボコンプレッサ吐出圧低下 (2) 助燃バーナ失火 D システム異常 (1) 計装空気圧力低下 (2) 1ループコントローラ異常 (3) 制御用計算機システムダウン E その他 (1) 可燃性ガス漏えい (2) 火 災 (3) 地 震	(1) 保護回路動作不能 (2) 制御電源そう失	A 燃料改質装置起動失敗 (1) 天然ガス圧力低下 (2) 水蒸気圧力低下 (3) 改質器バーナ点火失敗 (4) 冷却水ポンプ起動異常 (5) 反応器升温遅れ (6) 改質器炉内圧低下 B 燃料電池モジュール起動失敗 (1) 冷却系ポンプ起動異常 (2) 燃料電池予熱系異常 (3) 燃料電池モジュール升温遅れ (4) 燃料電池モジュール開放電圧異常 C ターボコンプレッサ起動失敗 (1) コンプレッサ起動失敗 (2) 助燃バーナ失火 D システム異常 (1) 起動用ボイラ異常 (2) 起動用ボイラ・バーナ失火 (3) 窒素タンク圧力低下 (4) 純水タンク液面低下 (5) 水素タンク圧力低下



な知見を得たので、今後の実施プラントの設計に反映させる。

6. む す び

50 kW-MK II プラントを例として、燃料電池の特長を最大限に発揮するために必要な制御システムの構成、機能及び運転結果について述べた。このMK II プラントで開発した制御システムにより、MW 級発電システムに要求される制御仕様を十分達成していることを示した。

MK II プラントでは、骨子となる制御方式は十分実績のある

PID 動作を採用した。更に将来のプラントでの制御システム設計には、負荷速応性のより一層の向上を目指し、多変数制御、適応制御などの手法も検討している。また計算機シミュレータを併せて開発し、動特性を含めて全プラント構成の詳細設計を進めている。

参 考 文 献

- (1) 岸田ほか：三菱電機技報，55，No. 7，P. 475 (昭56)
- (2) 岸田ほか：三菱電機技報，57，No. 5，P. 362 (昭58)

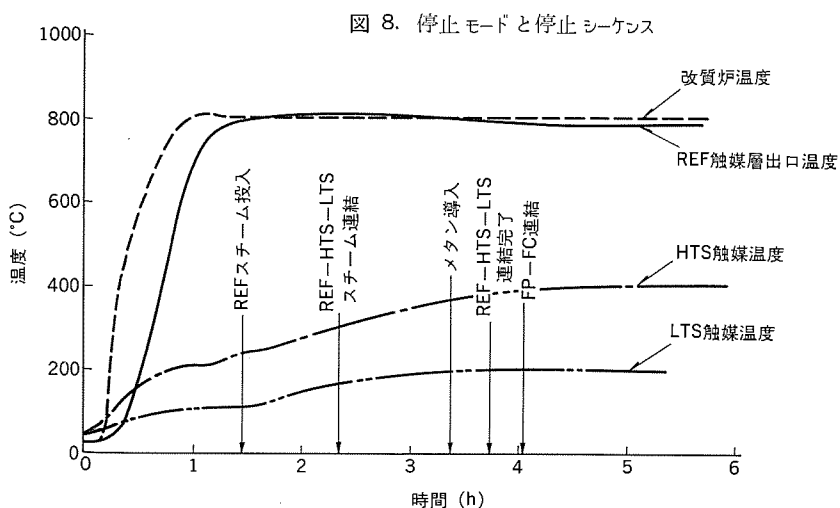


図 9. 50 kW-MK II プラントの起動特性

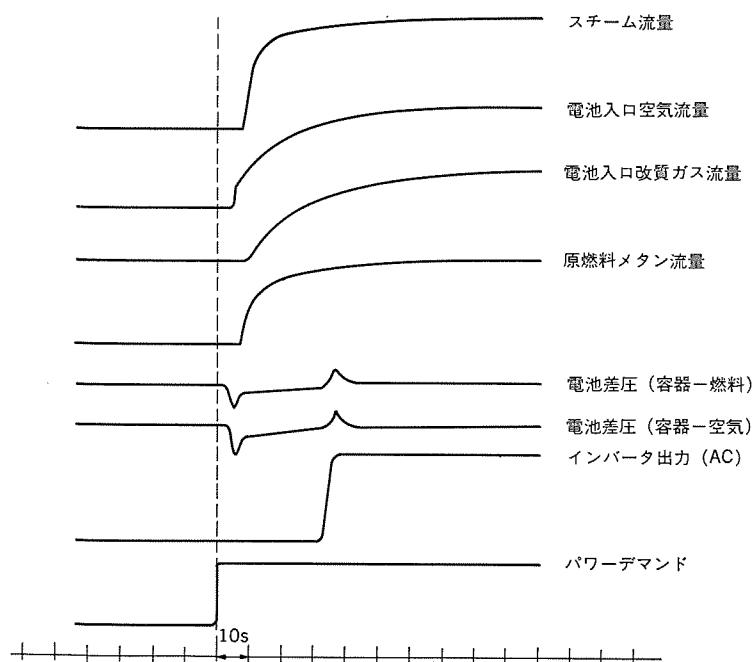


図 10. 50 kW-MK II プラントの負荷応答特性

火力発電プラントの予防保全システム《MELRAP-T》

高田 信治*・広崎 吉彦*・明神 俊治**・仁科 秀夫*・富田 義信***

1. ま え が き

火力発電プラントの中で、納入後15～20年を経過するものが多数出現しはじめており、設備を有効に活用してゆくために、予防保全の重要性が一段とクローズアップされる時代を迎えた。

当社では、これらの要望に対処するために、電算機を応用した予防保全システム《MELRAP-T》(Mitsubishi Electric Company Reliability & Availability Up-Grading Program-Thermal)の開発を行っていたが、昭和58年度にシステムを完成し実質的な活動に入った。

《MELRAP-T》システムは、各発電ユニットの主要構成電機品ごとに機器固有の予防保全項目を定め、予防保全項目ごとに実施された履歴をベースに、次回実施すべき予防保全項目と必要実施時期をデジタル化してアウトプットするものである。電算機を利用することにより、多数のユニット・機器に横断的予防保全項目を提供できるようになり、火力発電プラントの信頼性向上と稼働率向上に寄与できる基礎が完成したので、以下《MELRAP-T》システムの電算機フロー、運用方法、データ、アウトプットなどについて概要を紹介する。

2. システムの特長

《MELRAP-T》システムは、納入電機品を機種分類し、機種に必要な予防保全項目を定めて、発電ユニットごとの納入品リストをキーとして、ユニット別、納入機器別の予防保全項目をアウトプットすることを基本としており、次の特長をもっている。

- (1) 基本的予防保全項目が、機械的に電算機で処理されるので、多数の納入ユニット・機器に対して、横断的に落ちなく、予防保全項目をピックアップすることができる。
- (2) 機械的にピックアップされた予防保全項目に対して、個別に検討を加え、不要項目の削除、プラント固有条件の追加、個別の予防保全実施履歴のインプットにより、各プラントの納入電機品固有の予防保全項目、実施時期を提供できる。
- (3) 推奨項目には経年劣化だけではなく、定検時の重点点検、省力・自動化、運転中の重点監視、運用変化(DSS-Daily Start-up & Stop)などの項目が含まれているので、総合的なプラント改善資料として活用できる。
- (4) 予防保全実施履歴をアウトプットしているので、プラントの経歴表として活用できる。
- (5) 必要な予防保全項目を当年度だけでなく、時期を付して4年先までデジタル化しているので、長期的な予防保全計画ができる。
- (6) 予防保全項目ごとに提案書(1項目1葉)を作成し、更に必要なものには技術資料を添付し、これらの予防保全参考文献を体系化し、利用しやすいように配慮している。
- (7) 予防保全項目、点検・取替えの周期、診断基準などを、今後この《MELRAP-T》システムの中で、理論的裏付けと経験によってレビューしてゆくことにより、より高いレベルの予防保全技術へ発展

させることができる。

(8) 近い将来、当社の各製作所、支社(営業所)間をオンライン化するが、この時点で各端末からオンラインでこのシステムを呼び出すことができるように計画している。

(9) システムへのインプット、アウトプットは漢字処理しており、非常に見やすくなっている。

3. システムの概要

3.1 情報ルート

《MELRAP-T》予防保全システムの運営情報ルートを図1.に示す。ハードウェアを担当する各製作所は、次のベースデータを作り、技術取りまとめ部門(制御製作所火力プラント部)へ送付する。

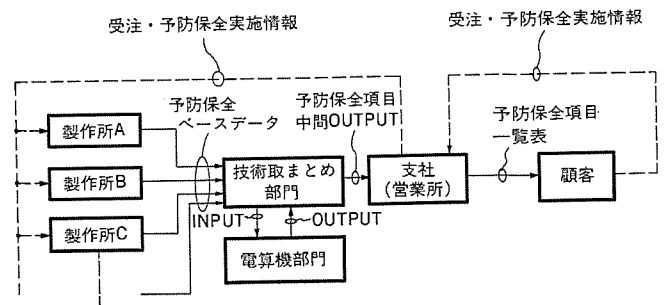


図1. 《MELRAP-T》予防保全システムの情報ルート

- (1) 各ユニットに納入された納入品リスト
- (2) 機種ごとの予防保全項目と点検・取替えなどの周期を定めた機種マスタ
- (3) ユニット個別の予防保全項目修正情報(削除・追加を含む)
- (4) 予防保全項目ごとの実施履歴情報

これらのデータは、電算機処理されて予防保全項目中間アウトプットを出力し、社内のチェック&レビューを受けて修正情報を再度電算機処理して、顧客に提出する予防保全項目一覧表をアウトプットする。予防保全項目一覧表にもとづいて、当社と顧客の間で技術説明を含めた打合せなどを行う。顧客から受注した案件と予防保全実施情報は、直接又は間接(支社経由)に製作所へ伝達され、これらの情報を次のサイクルの電算機処理に反映する。これらの処理サイクルは、原則として年1度としている。顧客へ提出する予防保全項目一覧表は、毎年7月を予定しており、次年度の予防保全計画に利用できるようなスケジュールとしている。もちろん、特別に必要なユニットについては、必要時に予防保全項目をアウトプットできるようにしている。

3.2 電算機処理フロー

《MELRAP-T》予防保全システムの電算機処理フロー概要を図2.に示す。以下図により説明する。

3.2.1 第1ステップ

ユニットマスタ(M1)には納入発電ユニットを、納入品リスト(M2)には各ユニットごとの納入電機品を、機種マスタ(M3)には電機品を

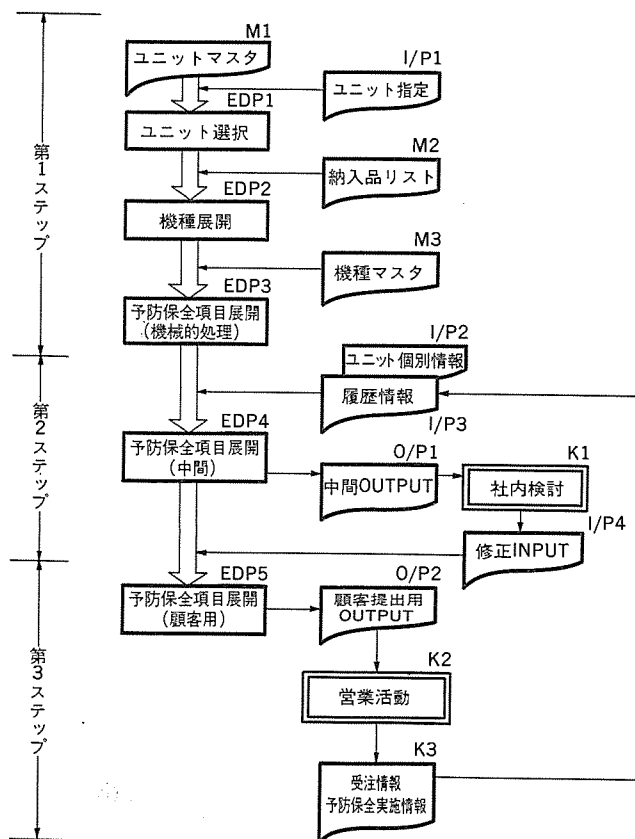


図 2. 《MELRAP-T》予防保全システムの電算機処理フロー

機種分類して機種ごとに定めた予防保全項目、点検・取替え周期などを登録しておく。

アウトプットを得たいユニットに対して、ユニット指定 (I/P1) を行えば、ユニットマスタ (M1) からユニット情報を取り出し、ユニット選択 (EDP1) が完了する。ユニットコードをキーとして、納入品リスト (M2) から対象ユニットの納入品を引き出し、機種展開 (EDP2) を行う。

対象機種すべてに対して、機種コードをキーとして、機種マスタ (M3) から機種ごとに予防保全項目展開 (EDP3) を行う。これで得られている予防保全項目展開 (EDP3) は、機械的に展開されたもので、ユニット個別のメニューとなっていない。

3.2.2 第2ステップ

第1ステップで得られた機械的処理展開を、ユニット個別のデータに変えるのが第2ステップである。

納入品個別を対象に、機種マスタの中の不必要項目をピックアップし、これらを取消し処置するとともに、ユニット固有条件を追加・修正する。すなわちユニット個別情報 (I/P2) 処理を行う。また、納入品の各予防保全項目ごとに、過去に予防保全が実施されたものに対して実施年月を、履歴情報 (I/P3) としてインプットする。

予防保全項目ごとの実施推奨時期は、過去の履歴情報 (I/P3) があるものはこれをキーとして、ないものは納入年月、又は運用開始年月をキーとして周期計算し、予防保全項目展開 (EDP4) の処理を行い、中間アウトプット (O/P1) を出力する。中間アウトプット (O/P1) をベースに、当社内の社内検討 (K1) を行い、修正 INPUT 処理 (I/P4) をする。

3.2.3 第3ステップ

修正 INPUT (I/P4) により、すべてのインプットが完了し、顧客

に提出できる予防保全項目展開 (EDP5) が完了し、顧客提出用 OUTPUT (O/P2) が得られる。これらをベースツールとして、技術説明会などを含めた営業活動 (K2) を行い、受注情報、予防保全実施情報 (K3) をシステムフィードバックする。

4. 対象プラントと対象機種

このシステムの対象プラントは、既設の発電プラントを対象としており、当社が主要機種を納入しているプラントについて、展開を図っている。主要対象機種と担当製作所について表1.に示すが、主要電機品は、ほぼすべてこのシステムの対象としている。

表 1. 対象機種と担当製作所

機	種	担当製作所
発 電 機	水素冷却機 水冷却機 空気冷却機	神 戸 製 作 所 長 崎 製 作 所
	ブラシレス励磁機 サイリスタ励磁器 直流励磁機	
	外鉄形油入変圧器 内鉄形油入変圧器 負荷時電圧調整器 乾式変圧器 分路リアクトル	赤 穂 製 作 所 伊 丹 製 作 所
変 圧 器		
回 転 機	高圧電動機 ボイラ用通風機	
開 閉 所 機 器	遮断器 断路器 GIS, GIB 壁ブッシング がいし形計器用変流器	伊 丹 製 作 所 丸 亀 製 作 所
	GMCS 界磁遮断器 気中遮断器 磁気遮断器 キュービクル メタクラ パワーセンター コントロールセンター バスダクト	伊 丹 製 作 所 丸 亀 製 作 所 名古屋製作所
制 御 盤 制御装置	BTG 盤, 保護リレー盤, etc. AVR ALR カバナ制御装置 タービン監視計器 タービン保安装置 無停電電源装置 制御盤	制 御 製 作 所
計 算 機	計算機	コンピュータシ ステム製作所

5. システムを支えるマスタ

インプットデータは、対象プラントのユニットごとの納入品情報、納入品に対する予防保全項目とその項目に対する履歴情報などで膨大な量となっている。インプットデータの代表的なものについて、以下に説明する。

5.1 ユニットマスタ

当社が主要対象機種を納入している対象プラントに対して、ユニット別にユニットコードを与えており、ユニットと納入品を結びつけるためのキーコードとしている。現在約400のユニットコードを登録している。

(1) ユニットコード

ユニットコードを図3.に示す。コード体系としては、今後火力発電以外

内容として修正管理フォローできるようにしている。すなわち、納入品固有の種々条件を加味して展開される。

図 4. に基づき予防保全項目一覧表について説明する。

(1) ユニットコード、ユニット名、機種名

図 4. の左上部にユニットコード及びユニット名を、また代表機種名欄に対象納入品の機種名をアウトプットしている。

(2) 機種名・オーダー No. 又は製造番号

この覧には納入品の用途・仕様、オーダー No. 又は製品番号、納入年をアウトプットしている。

(3) 保全コード

左の欄は、機種に対する予防保全項目の固有番号であり、ランニング番号で採番しており、2けた(桁)としている。右の欄は、予防保全項目の提案推奨理由として次のリストの中から1又は2項目を登録できるようにしている。

- | | |
|--------------|------------------|
| 1 : 事故未然防止 | 6 : 運転中の重点監視・改善 |
| 2 : 最新設計適用 | 7 : 定検作業改善 |
| 3 : 省力化、自動化 | 8 : 省エネ・高効率化 |
| 4 : 経年劣化、再生 | 9 : 運用変化対策 |
| 5 : 定検時の重点点検 | 0 : 生産中止に伴う代替品推奨 |

この推奨理由区分により、提案内容を分類することができ、アウトプットのみで内容を判断できるようにしている。

(4) 予防保全項目

機種マスタの機種コード分類にて一応細分化され、その機種に相当した予防保全項目となっているが、納入品固有の条件を加味し、予防保全項目の抹消、追加、修正などを行い、納入品独自の予防保全項目として運用する。

図 4. では機種マスタとして17個の保全コードをもっているが、07, 08, 13~16の項目はユニット固有の条件を加味して、抹消している。また、項目10, 11については推奨実施予定欄にマークを付してないが、この納入品のもつべき予防保全項目としてアウトプットしている。

(5) 周期

予防保全項目の内容により、1回実施すれば今後対策不要の項目、及び周期的に実施の必要な項目があり、前者には周期を記入していない。実施周期は、点検及び経年劣化・再生の予防保全項目が主であり、周期設定にあたっては余寿命判定評価、経年劣化管理、事故・故障の履歴、定期的検査など総合的見地から、予防保全項目ごとに設定している。納入品の特殊状態のものについては、納入品個々に周期の設定変更ができる。

(6) 製作期間

予防保全項目の実施に必要な工場での製作期間を示す。

(7) 現地工期

予防保全項目の実施に必要な現地での作業工事期間を示す。納入品の状態、据付け(取付け)状態などにより工期の変更もできる。

(8) 実施経歴

実施経歴は、当年度(太線の黒枠年度を示す)前3年間については

図 4. に示すように●, ■, ☆, ◆のマークで実施を表示している。また、当年度4年以前は実施年の表示としている。履歴情報は、このシステムの一つの大きな特徴であり、納入品の履歴管理と共に実施推奨時期の計算に使用する。履歴情報を漏れなく、きめ細かくインプットすることが、このシステム運用上不可欠であり、顧客の協力が必要である。

(9) 実施予定

予防保全項目の実施年度を当年度を含め4年間にわたり、推奨している。推奨レベルは、是非実施願いたい項目(◎印)、緊急を要しないが実施をお願いしたい項目(○印)のマーク付けとしており、また顧客の計画予定済み(※印)の項目についてもマーク付けとしている。マーク付けの基本は、実施周期のある項目は納入年、又は運開年から起算するが、実施済みのものは実施年から起算して周期計算を行っている。

なお、中間アウトプットでは、対策範囲、点検内容など作業効率を検討し、推奨年度の相違している項目群を同一推奨年度とするなどを考慮して、修正インプットを行う。

上述のように当年度を含め4年先までの実施推奨としており、長期的な予防保全計画ができるようにしている。

(10) 資料番号

予防保全項目に対する説明資料を示し、1項目1葉で提案内容の概要が分かるものとなっている。記載内容は提案内容、改善前内容、改善後内容、詳細説明資料番号を記している。

(11) 資料編集

《MELRAP-T》システム資料は、次のように編集している。

- ・予防保全項目一覧表(電算機からアウトプットしたもの)
- ・予防保全提案書
- ・詳細説明資料

これらの3分類で製本ファILINGして、ユーザーへ提出している。詳細説明資料は、予防保全提案書(一項目一葉)では、十分技術的に説明できないものについて、補足説明用として作成したものである。

7. む す び

以上、《MELRAP-T》予防保全システムの概要について述べた。今後、このシステムを有効に活用してゆくためには、顧客とメーカー間の技術交流、情報交換をより一層密として、

- (1) 新しい予防保全項目のピックアップ
- (2) 点検・取替周期の理論付け
- (3) 劣化診断技術のレベル向上

を進めて、これらをこのシステムに反映してゆく必要がある。

また、このシステムに蓄積されたデータの活用方法、他の分野への拡張なども今後の課題であり、今後共システムの改善に努力してゆく所存である。関係各位のより一層の御支援を、お願いする次第である。

姫路市水道局納め上水道配水コントロールシステム

駒田俊秀*・阿藤隆英**・半田 哲***・福島勇二+・中島弘善**

1. ま え が き

我が国の上水道の設備はほとんどの都市において普及し、浄水場、取水施設の合理化や、ポンプ場、配水池の無人化などの整備が進められつつある。特に最近是需要家末端水圧の安定化を図る水圧調整や需要予測に基づき、ポンプ、バルブの計画運転をねらった配水池運用など広域にわたって設置されている送配水施設を合理的、経済的に運用し、需要家に安定供給することが要請されている。

ここで紹介する姫路市配水コントロールシステムは、広域に散在する取水から給水までの水道施設を一元管理し、施設全体として計画的に効率よく運用するものである。

このシステムの特長は次のとおりである。

- (1) 需要予測による配水計画と実績値に基づく計画のオンライン修正により、通常運用や、緊急時の迅速な対応のガイダンス
- (2) 配水管網シミュレーションを用いた末端水圧調整ガイダンス
- (3) 管理センターにおけるプロセス情報や運用管理情報の一元管理、及び各サブセンターへの情報サービス
- (4) 将来の施設計画、配水計画の策定に有用な統計管理資料の作成

2. 配水制御施設の概要

姫路市の水道は、昭和4年に市水道が給水を開始して以来、人口増、生活水準の向上、産業の発展などにより現在上水道普及率98.6%、給水人口443,000人、1日最大給水量184,000m³/日に達している。昭和54年6月から自己水源の不足分を兵庫県水道用水供給事業から受水しており、今後の水需要の増加はこの県水で補われる。

自己水源を最大限に使い十分安定した給水状態に保つこと、自己水よりも単価の高い県水を最大効率で運用することが姫路市水道事業にとって重要な課題である。この配水コントロールシステムは、全水道施設の総合運営を図るため昭和48年から進めている第5次拡張事業の一環として55年8月に着手し、58年7月に完了した。

姫路市域には市川をはじめ、夢前川、揖保川の三つの河川が流れており、上水はこれらの河川を水源として取水している。そのため浄水場、配水池、ポンプ場などの水道施設が数多く広域にわたって散在している。配水コントロールシステムの導入により、これら施設全体の一元管理と施設全般の操業が最大の効率で運用できるよう、取水から給水までの配水系統の水量、水圧、水位、水質及び設備機器の監視、制御を行い水の安定供給を図ったものである。

施設の構成は図1.に示すように保城浄水管理センターに中央制御室を設置し、これと浄水場、水源池であるサブセンター6箇所、無人の水源池2箇所、配水池19箇所、ポンプ場7箇所及び配水管路の端末監視点21箇所を遠方監視制御装置により、日本電信電話公社線を使用して情報伝送を行っている。

サブセンターは、水源系の取水から配水までの浄水処理プロセスをセンターに伝送するとともに、センターから指示されたポンプ運用ガイダンスに従って浄水場の運転を行う。無人水源池、配水池、ポンプ場は各施設の流量、水位、水圧、水質などのプロセス量や機器の運転状態をセンターに伝送し、センターからポンプ、バルブなどの機器が遠方制御される。配水管路の端末監視点では、需要地点の水圧や水質をセンターに伝送する。

管理センターでは、これらサブセンター、無人施設、端末監視点から集められた情報の監視と、計算機を用いてこれらの情報から配水量



図1. 施設の構成

* 姫路市水道局 ** 三菱電機(株)制御製作所 *** 同制御製作所(工博)
+ 同コンピュータシステム製作所

(2) 取水可能量予測

(3) 配水量配分

(4) 送水ポンプ運転計画

3.3 日配水計画修正

(1) 時間配水量予測

計画配水量と実績配水量を時系列的に解析し、数時間後までの時間配水量を予測する。

(2) 送水ポンプ運転計画修正

予測配水量と計画配水量の差が大きい場合、送水ポンプを計画通り運転すれば配水池水位の予測値は上下限を越える。したがって、この場合には送水ポンプの運転計画を修正する必要がある。

(3) 制限配水量計算

漏水のため取水量が減少した場合、施設の故障により計画通り送水できない場合、あるいは配水量が異常に増加した場合には、配水池水位を予測する。その結果、配水量の制限が必要ならば、制限する時間と配水量をオペレータに知らせる。

3.4 配水制御ガイダンス

需要分布の変動などの原因で管網内の水圧変動が生じた場合、又は実績配水量が制限を越えた場合は、オペレータが必要に応じて CRT 画面と会話しながら配水コントロールを行う。配水状況の監視は随時なされる。

(1) 需要推定

管網内のすべての需要量を実測することは不可能なために、一部の管路で計測される水圧と流量及び過去の検針データに基づいて需要量を推定する。

(2) 管網シミュレーション

推定需要量を用いて管網シミュレーションを行えば、管網全体の水圧、流量、流速などが推定できる。また、管網シミュレーションは、バルブ及びポンプの制御効果の評価にも利用される。

(3) 水圧制御 ガイダンス

オペレータが水圧実測値又は管網シミュレーションの結果をもとに水圧制御の要否を判断する。制御が必要な場合は最適なバルブ開度を計算し、その制御効果を CRT 画面に表示する。

3. 配水運用管理システム

配水コントロールでは、需要予測に基づいて施設の運用計画を作成し、配水状態が適正化するようにポンプ及びバルブの運用制御がなされる。図2.は運用管理システムのソフトウェア構成を示したもので、各機能について説明する。

3.1 長期配水計画

年単位で将来の需要量を予測し、上水道の施設計画及び長期的な配水運用計画を行う。

(1) 長期需要予測

需要量に影響する要因（人口、水道普及率、水洗化普及率、工場出荷額など）で予測モデルを作成し、実績データ及び将来の施策に基づいて回帰モデルにより10年先までの年間需要量を予測する。

(2) 県水受水計画

県と年間契約をして受水する場合、契約条件と配水池などの施設の制約条件のもとで受水量を最少化することが望ましく、線形計画法^{*1}を用いて最適な契約受水量と配水量を計算する。

3.2 日配水計画

日単位で全体の日配水量を予測して配水池ごとの時間配水量に配分し、配水弁の開度設定と送水ポンプの運転計画を行う。

(1) 日配水量予測

配水量は気象条件の影響を受けやすいため、前日と前々日の配水量、曜日、天候、最高気温、最低気温を変数とした GMDH^{*2} モデルを作成し、オペレータが予想天候、予想最高気温、予想最低気温を入力し

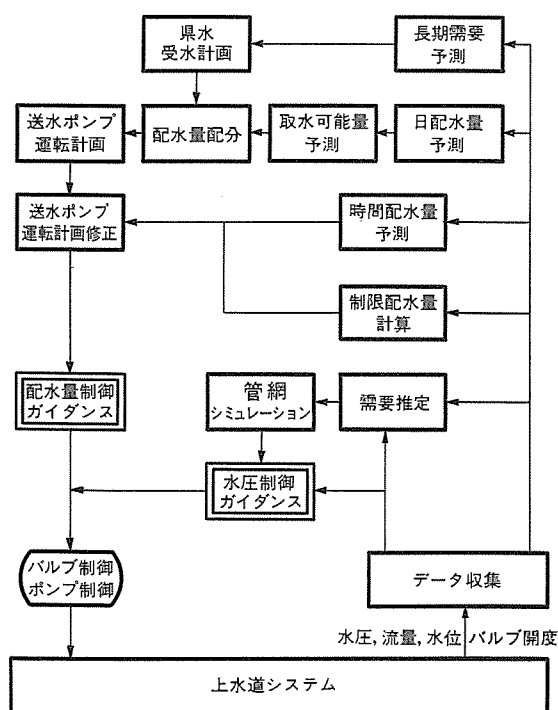


図 2. 運用管理システムのソフトウェア構成

*1 線形計画法：線形の等式及び不等式の条件のもとで線形の目的関数の極値を求める方法。

*2 GMDH: Group Method of Data Handling の略で、入出力データから自己発見的にモデルを非線形式で記述する方法。

*3 指数平滑法: t 時点において実績値 X_t が得られたとき, t 時点における期待値 S_t を, この X_t と $(t-1)$ 時点の期待値 S_{t-1} から求め, この期待値 S_t を用いて予測を行う方法。

(4) 配水量制御 ガイダンス

実績配水量が制限値を越えた場合には配水弁開度を計算し、オペレータに知らせて制御する。

3.5 データ収集

計算機には水系と管網における流量、水位、水圧、水質、バルブ開度、ポンプ運転台数などの膨大なデータが収集される。これらのデータは種々な処理をした後、監視盤及びCRT画面に表示され、オペレータがシステム状況の監視を行う。また、処理されたデータは日報、月報、年報、統計資料などの帳票の自動作成にも用いられ帳票の項目は気象条件、流量、ポンプ運転時間、薬品使用量、電力使用量などである。

3.6 その他

配水施設に事故などの異常が発生した場合、オペレータが迅速かつ的確に処置できるようCRT画面に事故復旧ガイダンスを表示する。ガイダンスの内容は、原因の推測方法、責任者への連絡方法、中央からのマニュアル運転方法などである。

4. ハードウェアシステム構成

配水コントロールのハードウェアシステムは図3.に示すように、各配水施設のプロセスを計測、制御する現場計装制御システム、これらのデータを伝送し、監視する遠方監視制御システム、管理センターの中央監視制御システム及び計算機システムから構成される。

4.1 現場計装制御システム

ポンプ場、配水池などの施設の現場には、配水コントロールに必要なプロセス量を計測するセンサーと管理センターからの指令に基づき、流量、

水圧、水位をマイナー制御するプロセスコントローラ《MACTUSシリーズ》、信号の中継、現場監視操作を行う計装制御盤などから構成される。

表1.にその機器と用途を示す。

4.2 遠方監視制御システム

このサブシステムは、全体システムの中ではデータ処理用計算機と各無人施設のプラント設備との間に位置し、監視制御情報の通信と非常時

表 1. 現場計装制御システム構成機器

分類	機 器	用 途
計 測	流 量 計	配水量、送水量、受水量、取水量計測
	水 位 計	配水池水位、浄水池水位計測
	開 度 計	バルブ開度計測
	圧 力 計	受水圧、配水圧、末端圧計測
	残 塩 計	〃
	pH 計	〃
	温 度 計	気象計測
監 視・制 御	プロセスコントローラ MACTUS 200	配水弁開度制御 配水流量パターン制御 配水池水位制御
	プロセスコントローラ MACTUS 610	送水ポンプ回転数制御
	計装・リレー盤	信号中継用バッファ、信号絶縁
	計装制御盤	現場機器の監視操作、指示記録

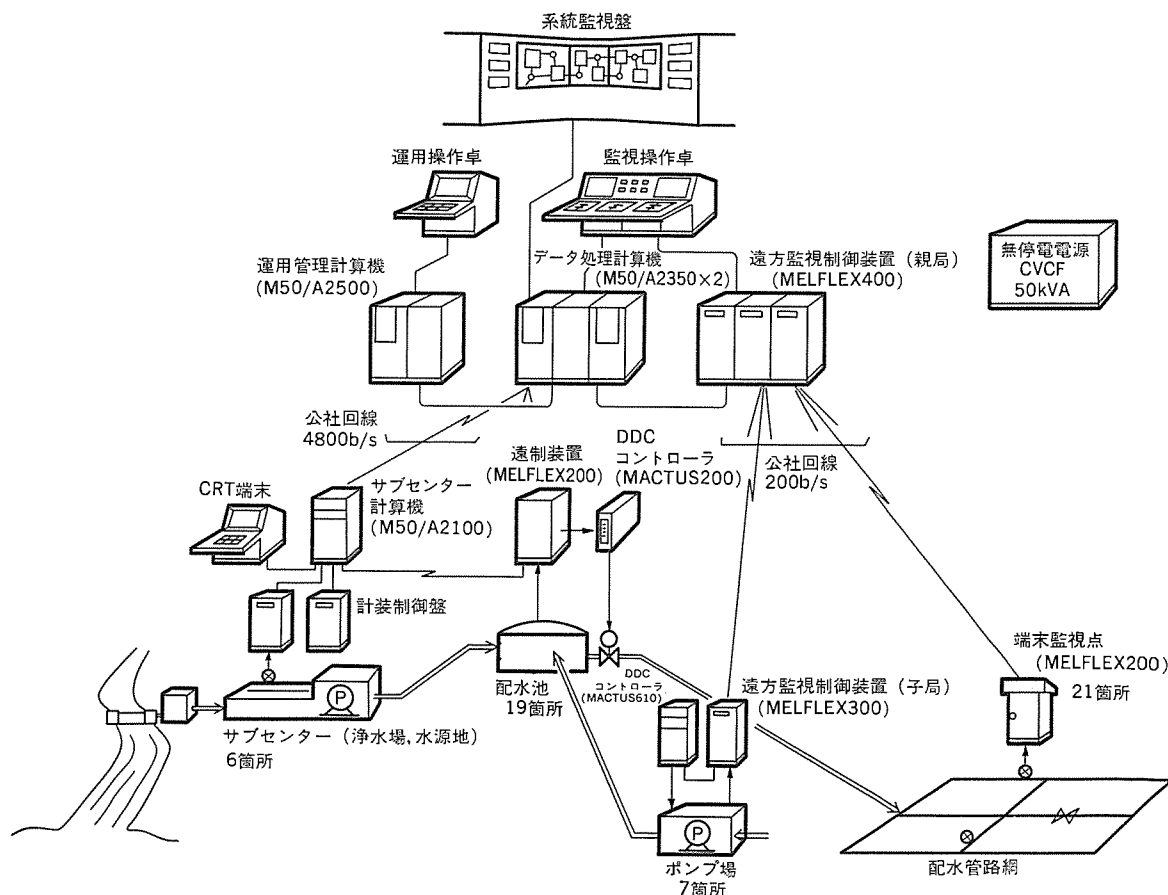


図 3. 配水コントロールシステム構成図

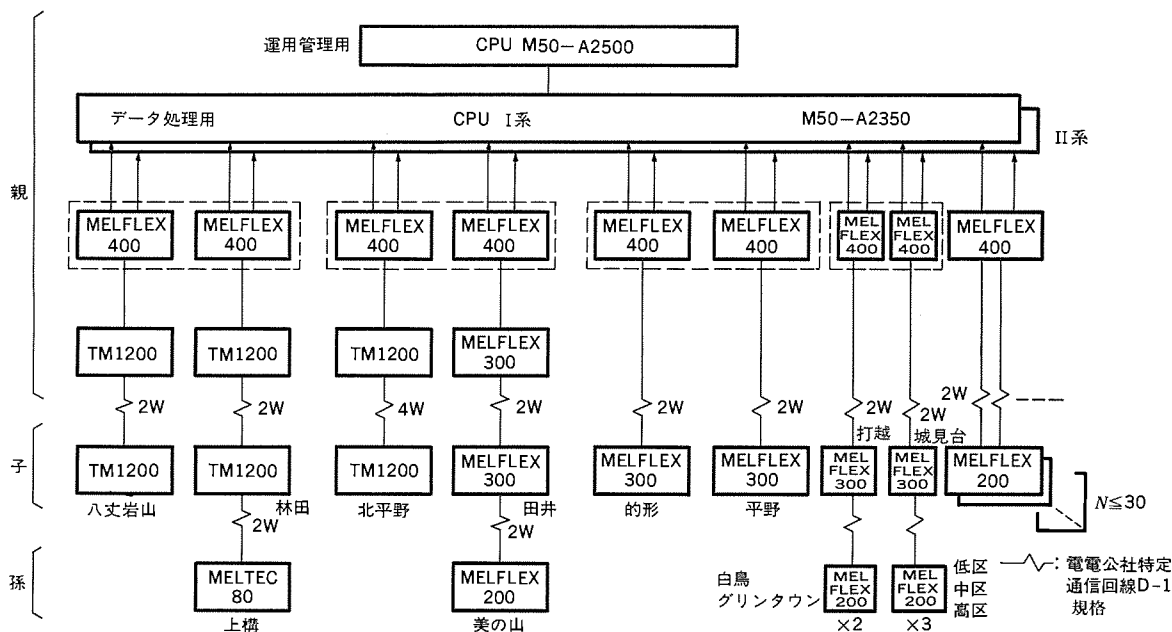


図 4. 遠方監視制御システム 系統図

表 2. システム仕様

(1)	伝送方式	(A) 親局—無人子局 子局—孫局 (1:1)×N 常時サイクリック形
(2)	通信回線	(B) 親局—配水管端末 1:N 回線選択形
(3)	伝送速度	日本電信電話公社特定通信回線 D-1 規格 2 線式 200 b/s
(4)	変調方式	周波数偏移 (FS) 変調
(5)	符号方式	NRZ 等長符号
(6)	符号化方式	(A) 反転二進送 (電気学会 44 ビット) 方式 (B) 隣接位相反転方式
(7)	同期	(A) フレーム同期 (B) ワード同期
(8)	誤り検定	(A) 反転二進送照合+パリティ検定 (B) 隣接位相反転照合+パリティ検定
(9)	伝送容量	最大 62 ワード
(10)	計測総合精度	±0.5% (常温常湿)
(11)	通話方式	交互使用

の通話を行う。その構成を図 4. に示す。保城センターにはデータ処理用計算機と高速にデータ伝送を行うため、《MELFLEX 400》を、比較的データ量の多い無人子局には《MELFLEX 300》を、またデータ量の少ない孫局や配水管端末には《MELFLEX 200》を配置し、経済的なシステムを構成している。また八丈岩山・林田・北平野・田井の 4 系統は、既に遠方監視制御装置が設置されていたのでこれはそのまま使用し、データ処理用計算機との間のインタフェース用に《MELFLEX 400》を設置した。

このサブシステムの概略仕様を表 2. に示す。親局と無人子局との間は障害が他系に影響しない 1:1 形で、また親局と配水管端末との間は、データ更新周期が長くてもよいので 1:N 形の対向とした。保城センターに設置される《MELFLEX 400》は上位のデータ処理用計算機の負荷を軽減するために、データのスケール変換や表示項目の状況変換などの一次処理を行い、DMA (Direct Memory Access) 転送により高速な伝送を実現している。

4.3 中央監視制御システム

広域にわたる施設の多量のデータを集中管理するシステムでは、施設の監視、操作、及び運用ガイダンスを操作員との対話により行うマンマシンインタフェースの設備は、誤りなく効率的に監視操作ができるよう考慮する必要がある。このシステムでは次のように機能的に使いわけている。

(1) システム監視盤

施設全体のマクロ把握を目的とし、地理的なイメージを加え、最小限必要の流量、水圧、水位の表示や、施設の一括故障表示などにとどめた。特にここでは計測信号の表示として液晶表示方式を採用した。これにより、アナログ表示ができること、上下限設定値と対比できることなど視覚判断の優れたものとした。

(2) 監視操作卓

施設の監視状況を詳細表示する CRT ディスプレイと、無人施設への機器操作、プロセスコントローラへの設定操作を行うパネル部から構成され、運転操作に必要な水源系ごとの施設のフロー表示、プロセスの異常、故障警報表示、トレンドグラフ表示及び各種データの表示を行う。また同一機能の操作卓 2 台により、2 系統への同時操作を可能とした。

(3) 運用操作卓

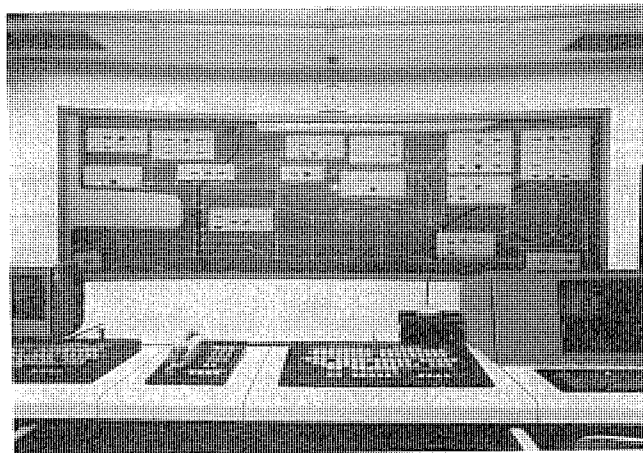


図 5. 管理センター中央制御室

運用管理用計算機専用のマンマシンインタフェースであり、配水量予測、ポンプ、バルブ運用計画、シミュレーションの実行結果などのガイダンス表示を行う。

(4) 記録装置

プロセスの監視操作状況を記録するアナウンスメントタイプライタ、日報作成用ロギングタイプライタ、月報、年報、各種統計資料を作成するラインプリンタ及びCRTハードコピー装置から構成される。(図5.)

4.4 計算機システム

配水コントロールシステムの中核となる計算機《MELCOM 350-50シリーズ》は図6.に示すように、運用管理(上位)、データ処理(中位)、サブセンター用(下位)とハイアラキ構成による機能分散を図り、処理性と信頼性を向上させた。

運用管理用計算機《MELCOM 350-50/A 2500》は配水予測、計画、修正、シミュレーション、統計処理などのオフライン的業務を主に行う。データ処理計算機《MELCOM 350-50/A2350》は、プロセスデータの収集、監視、制御データの出力、施設運転の状況、トレンドデータの表示、日報ロギング及び各サブセンターへのデータサービスなどのオンライン的基本業務を行う。特に信頼性を向上させるため、デュプレックス方式による二重化を図った。サブセンター用計算機《MELCOM 350-50/A 2100》は6箇所のサブセンターに設置され、センターのデータ処理計算機と公社回線(4,800 b/s(ビット/秒))により接続され、プロセスデータの送信、ガイダンス情報の受信とCRT表示などの業務を行う。

センターの上位、中位の計算機は、計算機結合装置(CLA)で接続するとともに、3台のいずれの計算機からも直接外部記憶装置をアクセスできる構成とし、計算機相互間の大量データの授受を可能としている。また中位計算機と遠制装置親局との間もCLA接続によりデータ転送の高速化を図った。

5. む す び

以上、配水施設の運用管理に計算機による管網シミュレーションをベースとした新しい手法を導入した姫路市配水コントロールシステムについて述べた。水道事業に求められている水資源の有効利用や、省エネルギー

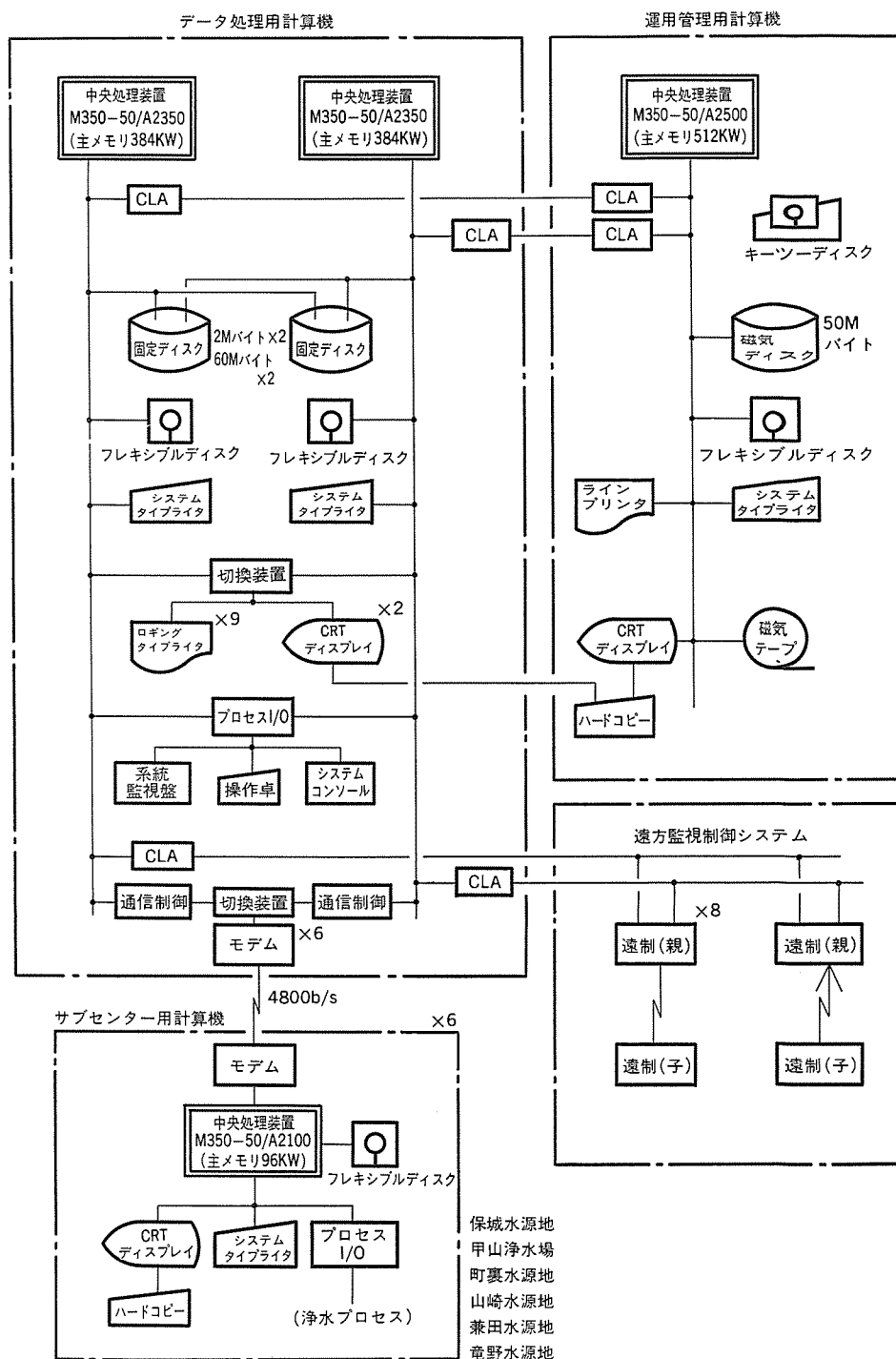


図6. 計算機システム構成図

ー化を図った効率の良い配水施設の運用に対し、このシステムはこたえたものといえる。今後更にデータの蓄積を図り、予測モデルやシミュレーションの精度を向上させ、最大限の能力を発揮させるための研究、検討を重ねていく予定である。

終りに、このシステムの設計から完成まで3か年にわたり御指導、御協力いただいた関係者各位に厚くお礼申し上げる。

新規格準拠三相誘導電動機《スーパーライン J,Kシリーズ》

柳内 芳彦*・前原 利昭*・鈴木 好弘*・立見 広光*

1. ま え が き

モートルは、設計技術・製造技術・材料技術などの発展と共に、絶えず進歩してきた。近年では、A種絶縁からE種絶縁を経て、今やB・F種絶縁を採用したモートルの時代へと移行しており、これら耐熱絶縁材料の採用による小形軽量化は世界的なすう勢にある。また、国際的問題の一つとして、技術的障害による貿易摩擦の回避が叫ばれており、日本の産業界は貿易摩擦回避の立場から国際規格との整合化を強く要請されることとなった。

これらの背景から、材料技術の進歩を取り入れ、IEC規格⁽¹⁾に準拠した最も合理化されたモートルの新規格が制定された。新規格は、JIS C 4210⁽²⁾ (0.2~37 kW) 及び JEM 1400⁽³⁾, JEM 1401⁽⁴⁾ (0.2~132 kW) で構成されている。絶縁階級は従来のE種に加えて、B種、F種を追加規定しており、E種は63~112 Mフレームに、B種は132 S~180 Mフレームに、そしてF種は180 Lフレーム以上に適用されている。また、国際規格との整合化の主眼である軸端寸法、 π - π 溝寸法公差は完全にISO規格、IEC規格に準拠している。当社では、これらの新規格に準拠したモートルとして新系列一般用三相誘導電動機《スーパーライン》Jシリーズ及びKシリーズを完成したので、その特長を以下に紹介する。(図1.)

(1) J, Kシリーズの特長

(a) 信頼性の向上

絶縁構成ではE種に加えて、B種、F種絶縁を採用し絶縁の耐熱性を向上した。また、軸受の潤滑には、特殊耐熱グリスを全シリーズに採用し、軸受の新形グリス排出装置の採用などとともに、全体的に信頼性の向上を図っている。

(b) 小形軽量化

小形モートルにおける鋼板フレームの拡大、小形中形モートルにおける通風冷却の改善、アルミダイカストファンを採用、中形モートルにおける絶縁グレードの向上による出力枠番適用の変更などにより小形軽量化を図っている。

(c) 作業性、保守の容易化

小形モートルでは電源との接続に、全面的に端子台方式を採用し接続作業を容易にするとともに、中形モートルでは、(a)項の新形グ

リス排出装置を、開放軸受に適用して保守の容易化を図っている。

(d) 特性の向上

小形鋼板モートルにおける効率の向上及び2極モートル全シリーズでのトルク性能の向上などにより、特性面でも向上を図っている。

(2) シリーズ範囲

シリーズ範囲は、枠番号63~315 Mフレーム、出力0.2~200 kW、の一般用低圧三相かご形誘導電動機シリーズである。

2. 信頼性の向上

新シリーズの絶縁は従来のE種に加えて、B種・F種絶縁を採用しているが、ここでは特に耐熱グレードの一番高いF種絶縁について説明する。また、モートルは回転機であるので、軸受潤滑性能は、機械としての信頼性に大きなウエートを占める。新シリーズでは、全面的に特殊耐熱潤滑グリスを適用しているため、この性能についても紹介する。

2.1 F種絶縁システム

耐熱絶縁システム⁽⁵⁾を採用したF種モートルは、当社では早くから多くの納入実績をもっている。その後、高分子材料を中心とした耐熱材料の開発や絶縁処理技術の開発導入により、F種モートルは当初の特殊専用の用途から最近では一般用として各種用途に適用拡大されつつある。新シリーズのF種絶縁システムは、①マグネットワイヤ、ワニス及び薄葉材料などの材料単体試験と、これらを組み合わせて行う適合性試験を含めたスクリーニング試験、②スクリーニングされた材料の一つの絶縁組織として組み立て、電動機絶縁としての機能を満足するかどうかを検討するための機能評価試験、③実際に使用される条件や環境を想定した実用性評価試験、などの過程を経て決定している。

この絶縁システムは、一般用途に使用されることから、特に耐環境性試験についても実用性試験のなかで重点的に評価を行い、開放形モートルでは、設置環境に存在するじんあいを模擬した人工じんあいをういて、これを吸湿させたステータコイルに均一に付着させ、引き続き40°C 95±5% RHの恒温恒湿槽にて吸湿し、相間、アース間に定格電圧を加えて長時間放置し、所定時間後の相間、アース間の絶縁抵抗を測定した。また、F種絶縁システムとしては、最も重要な耐熱寿命確認試験に対して、IEEE No. 117法に準じたリバーシング試験を行っている。巻線の温度上昇は正逆運転を行い、加熱、振動、吸湿、耐圧試験により評価している。

図2にこの結果を示す。図からもわかるように当社のF種絶縁システムは、十分な性能を持ち、実機における使用に際しても高い信頼性をもつことが確認できた。

2.2 潤滑グリス

回転機械であるモートルにとって、軸受性能の信頼性は重要である。軸受性能は軸受自身の信頼性と、更に潤滑による信頼性により大きく左右される。一般用の小中形モートルでは、ころがり軸受が使用されており、これらの潤滑にはグリスが用いられている。グリス潤滑に影響を及ぼす要因には、モートルの用途、運転条件、使用環境、駆

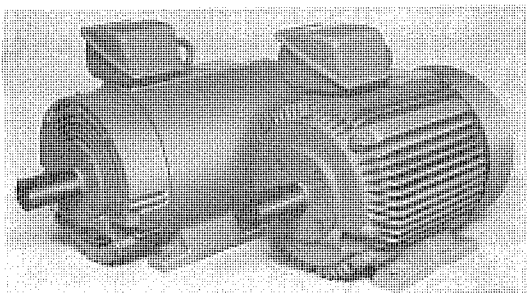


図1. 《スーパーラインKシリーズ》

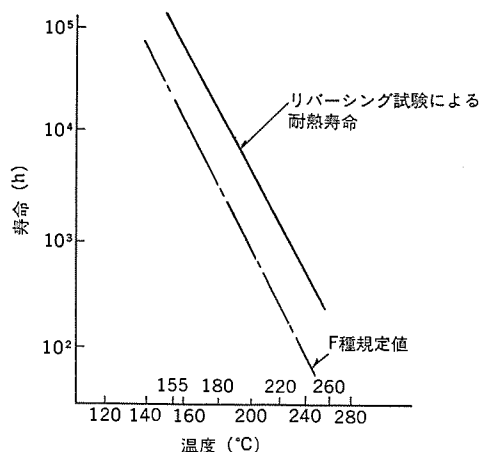


図 2. F 種絶縁システムの加速劣化試験による耐熱寿命

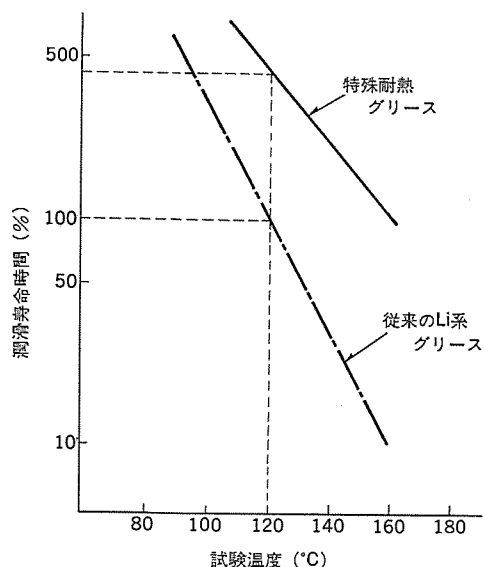


図 3. グリース耐熱寿命 (従来の Li 系 グリース 120°C の寿命を 100% としたときの寿命比較)

動方法、軸受種類などがあげられ、この要因に対してグリースに要求される性能が決定される。

グリースに要求される性能には、まず①使用温度範囲が広いこと、特に耐熱グレードが高いこと、②音響性能が良いこと、すなわち小形のモートルでは、軸受による音響が全体の騒音の中で大きなウエートを占めるので、軸受騒音を低減すること、③機械的に安定であること、これは密封軸受でのグリース漏れが少ないこと、更に混和安定性が良いこと、④防錆性能が良いこと、防錆としては、湿気、水及び塩水によるさび(錆)のほかに、モートル固有の検討項目として、コイルの絶縁ツニスに対する防錆性能を考慮する必要がある。⑤低温トルクが少ないこと、これは小形モートルでの機械的な摩擦損失を低減するためである。

以上これらの各種要求性能に対し、当社では 0.2 kW クラスの小形モートルから 200 kW クラスの中形モートルに適用可能な特殊耐熱グリースを採用した。図 3. はこのグリースの一特性を示すもので、使用温度に対するグリースの耐熱寿命を、従来のグリースと比較したデータである。このデータによれば、同一使用温度に対し従来のグリースに比べて約 4 倍の長寿命となり、回転機械信頼性向上に役立っていることがわかる。

また軸受グリース交換形の中形モートルにおいては、4 章で詳しく

説明する新形グリース排出装置⁷⁾を採用しており、グリース劣化後のグリース交換を、異常な温度上昇を招くことなく確実に実施可能としている。

3. 小形軽量化

小形軽量化は機械設計の根本的なテーマであり、回転電気機械では、従来から主として絶縁階級の向上によるところが大きい。新シリーズでは、中形モートル以上に F 種絶縁の採用拡大を行い、磁気回路設計の合理化、通風冷却性能の向上、部品単体材料の軽量化などにより小形化、軽量化を推進した。また小形モートルでは、更にフレーム材質を鋳物から鋼板化することによって 20~35% の重量低減、10~20% の容積低減を実現した。

3.1 鋼板フレーム化の拡大

小形モートルのフレームは、従来 63, 71 フレームに鋼板が採用されていたが、新シリーズでは、これを 80, 90 L フレームまで拡大した。フレーム材質を従来の鋳物から鋼板にすることは、フレーム材の重量だけをみれば、簡単に軽量化が可能と考えられるが、これを実現するためには、鋼板フレーム特有の設計技術、製造技術などの開発を必要とする。これら技術の中で、ここでは通風冷却の検討及び機械的強度の検討結果などについて述べる。(図 4.)

3.1.1 通風冷却

モートルの通風冷却は、機内に発生する損失をいかに効率よく外部に放散して温度上昇を低くすることができるかが最大の課題である。新シリーズには冷却方式として、直接外部の空気中で電気部分を冷却する防滴保護形と、内部の発熱を外被構造(フレーム、ブラケットなど)に伝え、これを外部の空気で間接的に冷却する全閉外扇形の 2 種類が

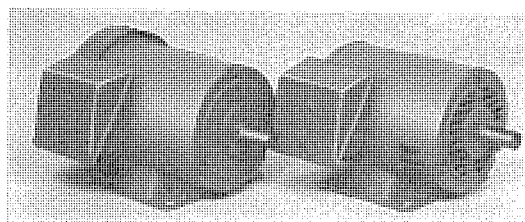


図 4. 鋼板フレームモートル (左: 全閉外扇形, 右: 防滴保護形)

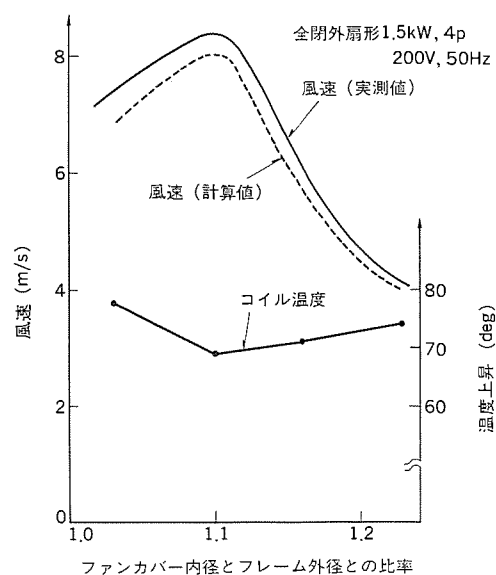


図 5. 全閉外扇形モートルの冷却効果

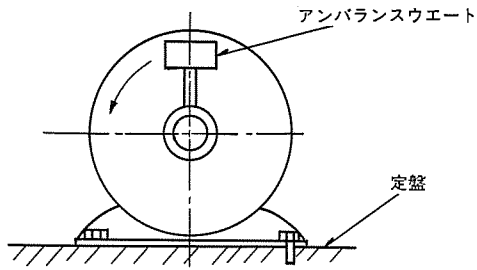


図 6. 機械強度試験方法

ある。鋼板フレームを採用するとき、通風冷却上、防滴保護形よりも全閉外扇形の方が技術的に難しい。

全閉外扇形の鋳物フレームでは、放熱効果を上げるために冷却フィンがフレーム外周に形成しているが、小形の鋼板フレームモートルでは、平板材を円筒状にしてフレームとするため、冷却フィンが設けられない。このためにフレーム外周の風速分布が変り、外部ファンによる冷却風が発散し、ファンカバー出口から遠ざかるにつれてフレーム表面の風速が低下してしまう。今回の鋼板フレームモートルでは、これを補い、更に冷却効果を上げるために、フレーム外径に対するファンカバー内径の比を変化させるとともに、冷却ファンの形状、相対寸法を変化させて試験を行い、通風冷却効果の良い構造を採用した。

図 5. は、フレーム外径に対するファンカバー内径の比を横軸にとりファンカバー出口の風速変化を表したもので、実機の巻線温度上昇及びフレーム温度上昇をプロットし、風速最大になるポイントが、最も冷却効果が良くこれにより適切なファンカバー形状を選定したものである。

3. 1. 2 機械的強度

小形鋼板フレームモートルの製造工程は、まず鋼板平板材を円筒状に曲げて継目を溶接し、鋼板製の取付足をフレームに溶接して、これに固定子を押し込み、かん（嵌）合部の機械加工を行ってからブラケットなどの部品を組み合わせて組立完了となる。したがってモートルとしての機械的強度としては、特にフレームと取付足の溶接強度が一番大きな課題である。

評価方法は、図 6. に示すように組立完了したモートルをベースに固定し、軸端にアンバランスウェイトを所定のラジアル荷重になるだけ取り付けて、これを 10^7 回以上加えて試験を行い、取付足の溶接部の状況を中心にセンターハイトの変化、軸振れの変化、エンドプレイの変化、ブラケットハウジングの変化などについても評価を行い、十分な強度をもつことを確認した。なお、当社ではフレームと取付足の溶接には、ロボットによる自動化を行っており、溶接強度も安定した性能をもっている。

小形鋼板フレームモートルにはフレームの鋼板化と共にブラケットについてもアルミダイカスト化を図り、フレーム、ブラケットなどの主要構造材料をすべて軽量化している。

3. 2 F 種モートル

2 章で説明したように、F 種絶縁システムを採用したモートルは、当社では以前から製作してきており、当初は特定の使用環境使用条件の用途に適用されていたが、次第に小形軽量化のメリットが認識され、現在では一般汎用の中形モートル（30 kW クラス）以上にすべて標準として F 種絶縁を採用している。

3. 2. 1 適用出力の変化

F 種モートルでは、従来の E 種モートルに比べて同一出力に対する枠番号の適用が、2～3 枠番（軸センターハイトでは、1～2 段）小さくなっている。これは高分子材料を中心とする耐熱絶縁材料の採用、

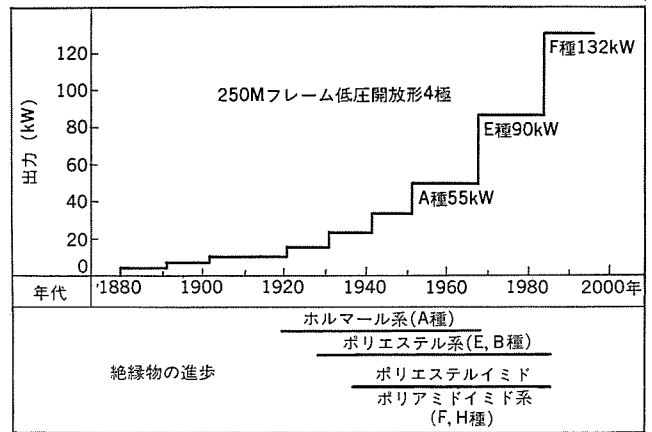


図 7. モートル同一フレームに入る最大出力の変化

通風冷却効果の向上及び電磁気設計の改善におうところが大きい。これらの中で、絶縁種別をベースに低圧開放形の 250 M フレーム 4 極に適用される定格出力の変化を振り返ってみると図 7. のとおりとなる。

すなわち、1950 年代、1960 年代のホルマール系材料を代表とする A 種絶縁では、55 kW であったものが、1970 年代から 1980 年代にかけてのポリエステル系材料を代表とする E 種絶縁では、90 kW となり、更に今回の F 種絶縁では、ポリエステルイミドやポリアミドイミド系の高分子材料の採用により、適用出力は 132 kW となっており A 種時代に比べて定格出力は実に 2.4 倍になっている。

3. 2. 2 磁気回路設計

F 種モートルにおける小形軽量化、すなわち枠番低下は、耐熱絶縁システムの採用だけでは達成できない。絶縁だけでなく、通風冷却の向上及び電磁気設計の改善も図る必要がある。

磁気回路設計においては、今回、回転電機設計の基本に立ち返り、電気装荷と磁気装荷の配分の合理化についても定量的な検討を重ね、適切な鉄心寸法を採用した。電気装荷であるアンペアコンダクタ AC と磁気装荷である励磁磁束 Φ は、定格出力 kW に対し次式で表される⁽⁹⁾。

$$(kW/p) / (\eta \cdot pf \cdot F/100) = K \cdot \Phi \cdot AC \quad \text{..... (1)}$$

ここで、 p : 極数、 η : 効率、 pf : 力率、 f : 周波数 (Hz)、 K : 定数

ところで励磁磁束 Φ は、空けきの磁束密度 B_g と空けきの面積 πDL の積で表される。また、アンペアコンダクタ AC は、導体の電流密度 ac と導体の面積 A_r 及び導体数 N で表される。そこで式 (1) の右辺は、磁束密度 B_g と電流密度 ac の積に比例することになる。

$$K \cdot \Phi \cdot AC = K_g \cdot B_g \cdot ac \quad \text{..... (2)}$$

空けき磁束密度 B_g に対する鉄心歯部磁束密度 B_t や、鉄心背部磁束密度 B_c の割合は、従来経験的に一定の割合を保っていた。このために空けきの磁束密度 B_g を大きくすることは、鉄心歯部及び背部の磁気飽和特性により上限があり、小形化することが困難であった。今回の電磁気設計では、この制限を取り払うために理論的な解析を実施して、磁束密度の磁路における配分を根本的に改良したものである。まず、鉄心歯部及び背部における最も合理的な磁気特性を、図 8. に示す磁界解析を用いて検討して、スロット形状と共に空けきを含めた磁束密度の配分を決定した。この結果、空けきの磁束密度 B_g は、従来に比べて 10～50% 大きくすることが可能となった。

ここで、式 (2) の空けき磁束密度 B_g と導体電流密度 ac をもとに、

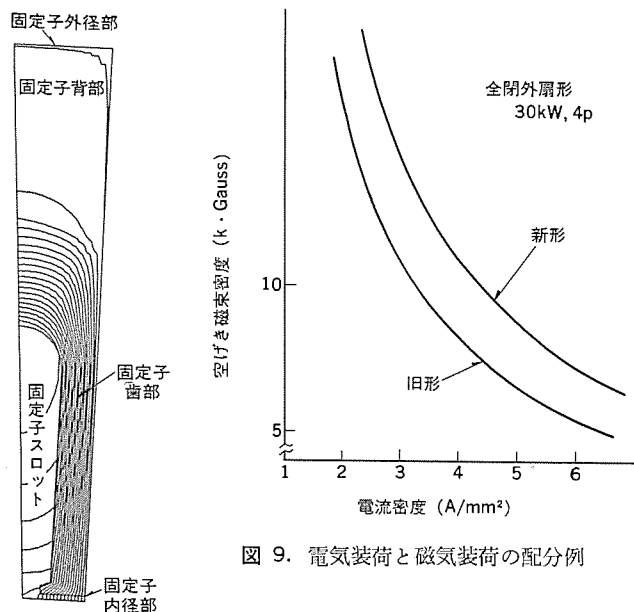


図 9. 電気装荷と磁気装荷の配分例

図 8. 固定子スロット磁界解析の例

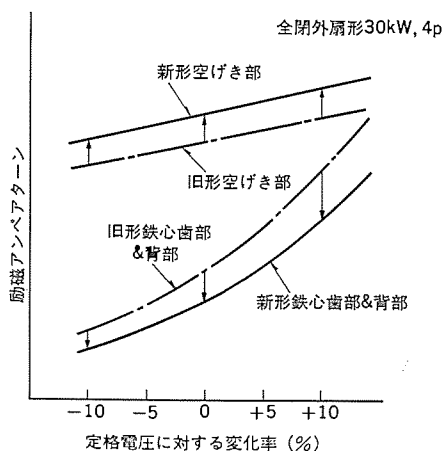


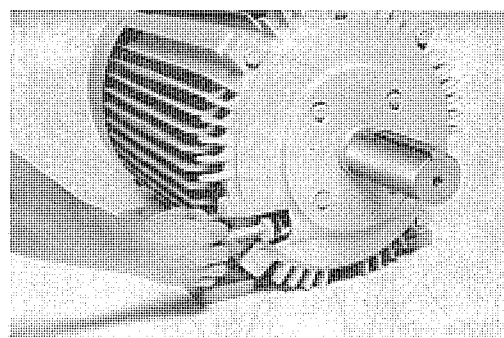
図 10. 電圧対励磁アンペアターン

新形及び旧形のカーブを描いてみると図 9. の例のとおり、 B_0 及び ac 双方の向上により、式(2)の定数 K_0 が低減され、全体的に 20～30% の小形化が可能となった。一方モートルに要求される項目の一つに適用性の拡大があるが、電源電圧の変動に対しても使用範囲が広いことが必要である。特に、電圧が高くなったときの磁気飽和が少なくことが望ましい。図 10. は、新形モートルが旧形モートルに比べて、使用電圧変化に対する磁気飽和が少なく使用範囲が広いことを示すデータの一例である。すなわち、新形モートルでは、鉄心部分の磁束密度を抑え、空けきでのアンペアターン消費を大きくとって磁気飽和を改善したものである。

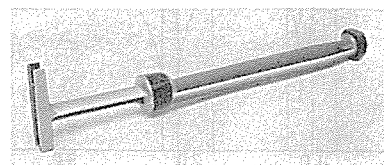
4. 取扱・保守の容易化—新形グリース排出装置

モートルの軸受潤滑については 2.2 節でも触れたが、グリース交換形の軸受を使用している中形モートルでは、設置場所における保守点検が容易にできなければならない。特に負荷機械側の軸受グリース排出は、出力軸端に取り付けられているラブリヤ、カップリングなどにより、作業スペースが少なく、安全な作業が困難なことが多い。

今回の新シリーズではこれら条件を改善させるために、従来と発想を変え、図 11. の新形グリース排出装置をグリース交換形モートルに全



(a)



(b)

図 11. 新形グリース排出装置

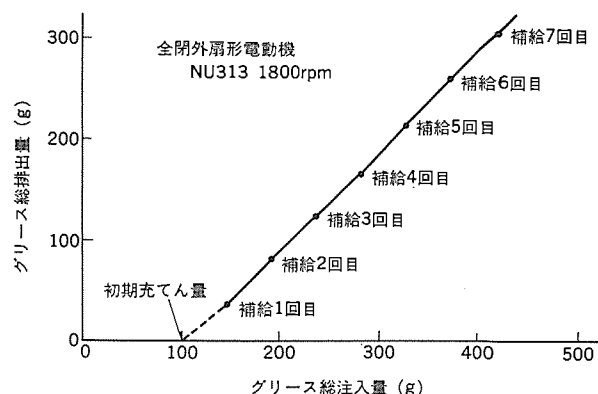


図 12. グリース総注入量と総排出量の関係

面的に導入した。この新形グリース排出装置は、次のような特長をっており、保守、点検は、安全かつ容易に行うことが可能になった。特長は、①負荷機械とのセットによっても影響を受けない。②排出グリースの処理が、確実かつ簡単にできる。③グリース排出装置の引出し方向は、モートルの左右いずれの方向でも可能である。④給油による温度上昇が小さい。⑤軸貫通部からのグリース漏れがない。⑥グリース排出経路が、軸受から最短のため新旧グリースの交換及び排出機能を十分に発揮することができる。

図 12. は、グリース総注入量と総排出量の関係を示すデータであり、確実な機能をもつことが分かる。

5. 特性—電磁騒音

今回の新シリーズでは、効率、トルク特性、電磁騒音などの特性面でも改善を図った。効率では、特に鋼板フレームモートルにおいて、また、トルク特性では、特に 2 極シリーズにおいて、固定子、回転子の溝数組合せをはじめとして、ロータの溝形状の工夫、特殊なエンドリングの導入を行い、フルアップトルクはもちろんのこと始動トルク特性を従来以上に改善した。またモートルの低騒音化への要求も増加の一途をたっており、これに対応するために電磁騒音の低減も行なった。ここでは紙面の都合上、電磁騒音の低減に限定して説明する。

モートルの電磁騒音の発生メカニズムは、一般に電磁加振力による強

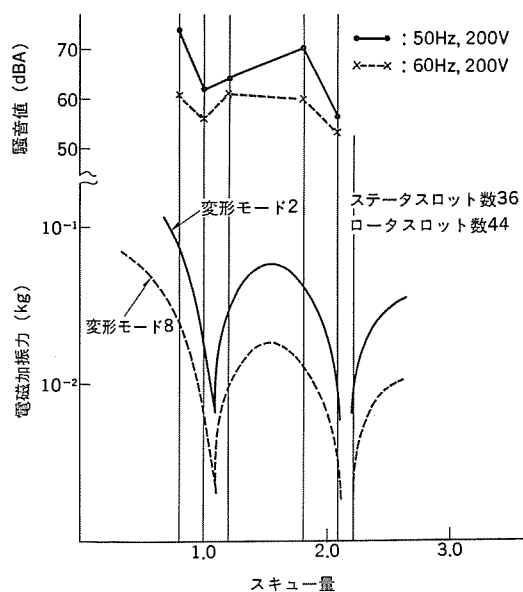


図 13. スキュー 変化による電磁加振力と騒音値 (0.75 kW, 6 p, 90 フレーム)

表 1. 力波のモードと周波数

na'	ma'	nb'	k	fk (Hz)	
				50 Hz	60 Hz
39 (Ns+P)	5 (Nr-Ns-P)	3 (P)	8	733	880
			Nr-Ns	$\frac{Nr}{P}(1-S)f$	
			2	633	760
			Nr-Ns-2P	$\left[\frac{Nr}{P}(1-S)-2\right]f$	
33 (Ns-P)	11 (Nr-Ns+P)	3 (P)	8	733	880
			Nr-Ns	$\frac{Ns}{P}(1-S)f$	
			14	833	1,000
			Nr-Ns+2P	$\left[\frac{Nr}{P}(1-S)+2\right]f$	

注 Ns: ステータスロット数 36, Nr: ロータスロット数, P: 極対数, S: すべり, f: 電源周波数

制振動によってモータの構造部材が振動し、騒音となって放射される。ここでモータの固有振動数が強制振動力の周波数と一致する場合は、モータが大きく振動し、騒音も大きくなる。更にモータの剛性が比較的弱い場合には、強制振動のピーク周波数が比較的大きく現れる。一般に音として問題となる力波の最大値は次式⁽⁶⁾で与えられる。

$$F_k = 6.48 \times 10^{-6} \frac{D \cdot l}{(g_e \cdot K_s)^2} \cdot (3N_1 I_1)^2 \cdot c \cdot A_k \quad \dots\dots\dots (3)$$

ここで、D: 空げき長, l: 鉄心幅, g_e : 有効空げき長, N_1 : 直列導体数, I_1 : 一次電流, K_s : 飽和係数, F_k : 力波, c: 係数, A_k : 力波のモードにより決まる定数

上式に基づいて、比較的磁気音の発生しやすい 6 極機についての計算値と実測値を次に示す。

表 1. は、力波の極対数と周波数についての計算値であるが、 $k=2$ は、固定子の変位がな円モードであり最も振動しやすい。図 13. は、ロータのスキュー量を変化させたときの力波の計算値 (図の

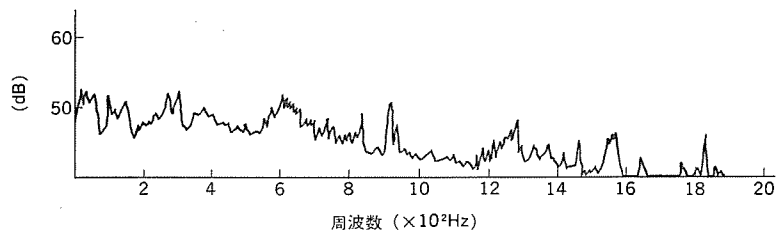


図 14. 騒音分析 (0.75 kW, 6 p, スキュー 2.1, 200 V, 50 Hz, 反端子側 0.5 m)

下部) に対し、実機の騒音レベル (図の上部) を比較したものである。これによってスキューがロータスロットピッチの 2.1 のときが最も騒音が低いことが分かる。図 14. は、ロータスキュー 2.1 の FFT による騒音周波数分析結果であり、良好な特性を示している。

6. む す び

今回の新系列モータは、国際規格との整合化を図った国内規格改訂に合わせて完成したものである。一般用モータの市場動向は、現在大きく変化しつつある。例えば、汎用インバータの出現により、従来一定速度であった一般用モータで、簡単に可変速運転が可能になるなど新しい応用分野が拡大している。また産業構造の変化により、小形で精密な、しかも制御可能なモータの需要増大などを筆頭に、モータは、より専用化されるとともに、高温使用や低温使用などの使用範囲拡大、更により長寿命、高信頼性の要求などが従来より増加していくものと思われる。

技術革新の著しい昨今、絶縁材料、構造材料、磁性材料、半導体材料、導電材料、製造技術などの進歩により、モータも更に進歩して行くものと考えられ、今後いっそうの研究を続け、より経済的で信頼性の高い電動機を製作するよう努力していきたい。以上のとおり電動機規格の改訂に伴った新シリーズの紹介をしたが需要家各位のお役にたてば幸いである。

参 考 文 献

- (1) INTERNATIONAL ELECTROTECHICAL COMMISSION (34-1, 34-9, 34-12, 72)
- (2) JIS C 4210-1983, 一般用低圧三相かご形誘導電動機
- (3) JEM 1400-1983, 一般用低圧三相かご形誘導電動機の寸法
- (4) JEM 1401-1983, 一般用フラジ形低圧三相かご形誘導電動機の寸法
- (5) INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO R 773)
- (6) 脇坂ほか: 低圧誘導電動機の F 種絶縁, 三菱電機技報, 48, No. 3 (昭 49)
- (7) 田仲ほか: 新給排油構造電動機, 三菱電機技報, 52, No. 10 (昭 53)
- (8) 竹内: 電気機器設計学, オーム社書店
- (9) 関野ほか: かご形誘導電動機の溝数組合せとスキューが磁気騒音に及ぼす影響, 三菱電機技報, 31, No. 4 (昭 32)

三菱オフィスターミナル M5000シリーズ

塚本 久雄*・守川 修平*・渡辺 透*・田中 雄三*・佐立 良夫*

1. ま え が き

コンピュータによる情報処理システムの構築方法には、次の2種類の流れが従来から存在している。

- (1) ホストコンピュータを中心としてトップダウンによるシステム構築
- (2) オフィスコンピュータを中心として、より現場に密着したボトムアップによるシステム構築

一方、マイクロプロセッサの急速な進歩と共に、パーソナルコンピュータを中心に、次のような非定型業務の事務処理の効率を図るオフィスオートメーション(OA)システム構築の流れが出現してきた。

- (1) 従来、手書き中心であった各種文書の作成にワードプロセッサを使用して効率をあげる。
- (2) 表形式の計算シートプログラムの利用により、非定型な事務計算処理の能率をあげる。
- (3) 文字情報に加えてイメージ図形情報を取り扱い、情報の表現効果や一覧性を高める。



図 1. オフィスターミナル M5000 モデル 30S

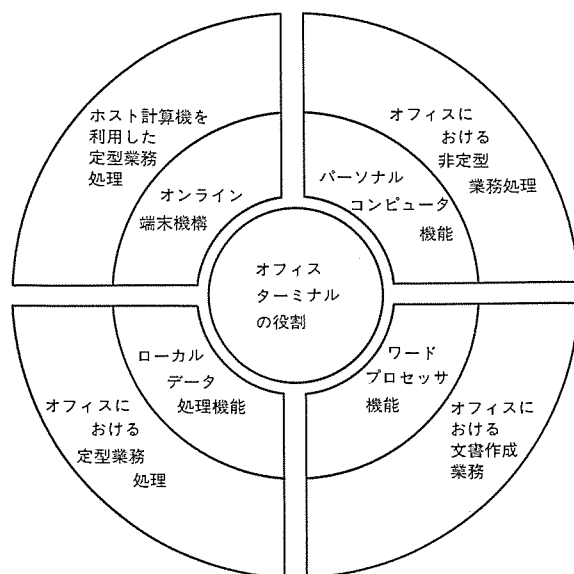


図 2. オフィスターミナルの役割

これらの処理は、従来のコンピュータシステム(ホストコンピュータ+従来端末)では、実施することが難しかったものであり、キーボードや表示画面を自由に操れる端末上でこそ比較的容易に実現できるものである。

オフィスターミナルは、従来からの情報処理システム構築方法の流れと、最近のオフィスオートメーションシステム構築の流れを統合化し、柔軟性に富む、使いやすいシステムを構築できるようにすることを目的として開発された新しい端末システムである。図2.にオフィスターミナルの役割をまとめて示す。

2. オフィスターミナルの利用形態と特長

2.1 オンライン端末として

オフィスターミナルは、従来端末(M4378, M4374など)と互換性を考慮して作られており、上位プロセッサ(ホストコンピュータやオフィスコンピュータ)からディスプレイ装置や日本語プリンタ装置に対し、従来端末と全く同じ方法でアクセスできる。

2.2 オフィスにおける定型業務のために

各部門における定型業務処理のために、高水準の機能をサポートするLEVEL II COBOL™, データエントリ業務に適したDD/SIMPE, 事務用BASICであるBI-286™などの言語プロセッサを利用して、オフィスターミナルのローカルデータ処理を行えるようにしている。

2.3 オフィスにおける非定型業務のために

MBASIC™や作表形簡易言語Super Calc™などを使用して、オフィスターミナルをパーソナルコンピュータとして利用できる。また、MELGRAPHやMELDRAWを用いて対話形式で各種のビジネスグラフや地図などの図形が作成できる。

2.4 オフィスにおける文書作成業務のために

日本語ワードプロセッサMELWORD-Jと英文ワードプロセッサWord Star™, Spell Star™及びMail Merge™を利用して文書作成ができる。日本語プリンタは、全角文字で24×24ドット、半角文字で19×11ドットの高品質なものであり、けい(野)線プリントも可能なのできれいな文書を印刷できる。また、2.3節で作成した図形データや作表データを変換ユーティリティを介して日本語文書に組み込むことも可能となっている。

以上の利用形態における各種適用業務プログラムは、すべてオフィスターミナルの基本ソフトウェアである拡張日本語MP/M-86™上で実行するので、業務を変える都度に初期プログラムロード(IPL)を必要とするようなわずらわしさは不要である。更に拡張日本語MP/M-86™のマルチジョブ機能を利用して、フォアグラウンド/バックグラウンドジョブで異なった利用形態のプログラムを同時に実行させ、必要に応じてワンタッチで新しい利用形態に切り換えることも可能である。

3. オフィスターミナルのハードウェア

オフィスターミナルには、シングルターミナルシステムのモデル10Sからマルチターミナルシステムのモデル50まで5モデルが用意されている。各モデルのハードウェアは、従来のオフィスコンピュータで培われた技術、パーソナルコンピュ

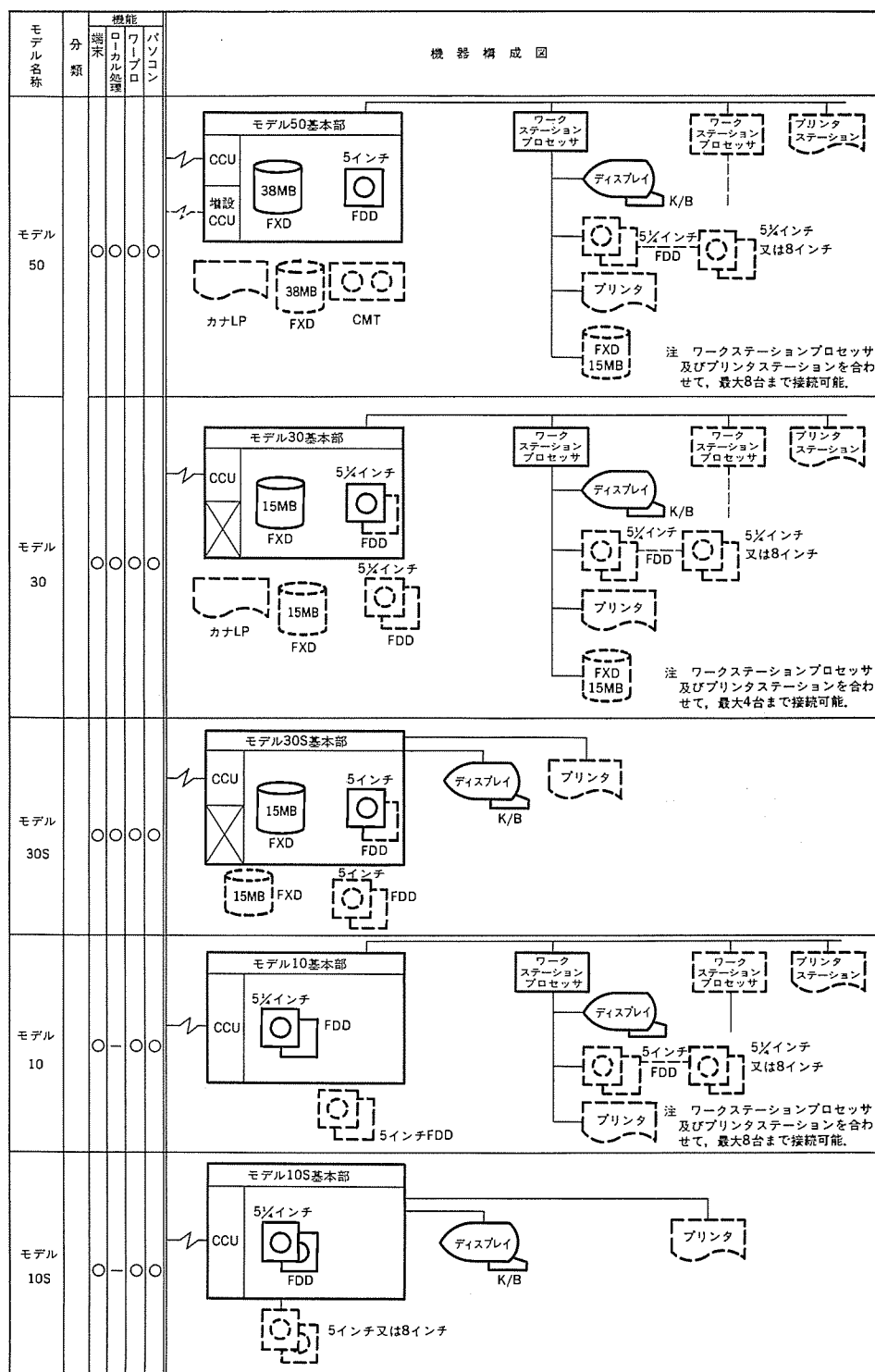


図 3. オフィスターミナルのモデル構成図

ータ、ワードプロセッサ、端末システムの技術を大幅に取り入れて開発されている。各モデルはモデル間で完全にアーキテクチャを統一し、シリーズ化が図られている。

オフィスターミナル各モデルの基本仕様を表1に示す。また各モデルごとの構成を図3に示す。図中点線で示す部分はオプションであり、最少構成から多種多様な大規模アプリケーションシステムまでの構築が可能である。次のこのシリーズのハードウェアの特長を述べる。

3.1 ハードウェアアーキテクチャの統一

シングルターミナルシステムのモデルからマルチターミナルシステムのモデルまで幅広いモデルを、標準コンポーネントを組み合わせることで構成できる。これにより各種適用システムの多様なニーズに対応できる。

3.2 マルチプロセッサアーキテクチャ

ワークステーションはすべてマイクロプロセッサを内蔵している。CCU(通信制御装置)とLNC(LAN制御装置)は、独立したマイクロプロセッサを内蔵して通信プロトコルを管理しているので、ワークステーションの処理性能を損なうことなく、ネットワークを構築できる。

3.3 マルチスクリーンのディスプレイ

フォアグラウンドプログラムとバックグラウンドプログラム用の各々に、コンソール画面とディスプレイ画面を各1面ずつ、合計4画面を独立に備えており、ジョブごとに切替表示が可能である。

3.4 高速グラフ描画

グラフ表示に専用LSIプロセッサを使用しており、高速図形描画が可能である。文字コード画面とは独立な画面(1,120×700)を3面(RGB対応)持ち、文字コード画面と重畳表示する機能を備えている。

3.5 エルゴノミックスデザイン

(1) 薄形キーボード設計

JIS配列キーボードは薄形に設計しており、更に6°、15°の傾斜角度調節ができオペレータの好みに合わせられる。

(2) ディスプレイのチルト機構

オペレータがみやすい位置に画面を設定できるよう、垂直方向5°~15°、左右方向30°の画面角度調節が可能なチルト機構を採用している。

(3) 目にやさしい、鮮明な高解像度画面

画面を反射防止加工し、照明や外光などの画面への写りこみを防い

でいる。更に1,120ドット×700ドットの高解像度画面を採用しており、日本語やグラフも、くっきり鮮明である。

(4) コンパクトな12形ディスプレイも可能

12インチディスプレイでの24ドット×24ドットの漢字表示を可能としている。

3.6 RAS機能

各制御部のパリティチェック機構、ディスク制御部のECC機構、バスタイムアウト、IPL時のベーシックアシユアランステスト(BAT)、パワーフューリア検知などを実行しており、中規模程度のコンピュータに匹敵するRAS機能をもっている。

3.7 豊富なディスクレパートリ

5 1/4インチの1Mバイトのフレキシブルディスク装置及び8インチの1Mバイトのフレキシブルディスク装置、15Mバイトの5 1/4固定ディスク装置、38Mバイトの8インチ固定ディスク装置と豊富なディスクレパートリを揃えている。

3.8 充実したプリンタレパートリ

日本語プリンタ装置、水平インサータ付き日本語プリンタ装置、日本語ラインプリンタ装置、オフィス環境に適合する静かな印刷ができるカラーサermalプリンタ装置など豊富な機種から選択可能としている。

4. オフィスターミナルの基本ソフトウェア

オフィスターミナルの基本ソフトウェアは、拡張日本語MP/M-86™を採用している。拡張日本語MP/M-86™は、米国デジタル・リサーチ社で開発されたMP/M-86™をベースに、日本語処理機能、ファイル管理機能、画面管理機能、ジョブ管理機能、ローカルエリアネットワーク(LAN)管理機能、オンライン端末機能などを強化するとともに、性能向上を図った基本ソフトウェアである。

また、MP/M-86™はパーソナルコンピュータの基本ソフトウェアCP/M-86™と上位互換性をもっているため、パーソナルコンピュータのCP/M-86™上で動く各種流通ソフトウェアの利用が可能である。拡張日本語MP/M-86™には、次の2種類を用意している。図4にオフィスターミナルの基本ソフトウェア構成を示す。

(1) システム1

表1. オフィスターミナルの基本仕様

モ デ ル	M 5000 モデル 10 S	M 5000 モデル 10	M 5000 モデル 30 S	M 5000 モデル 30	M 5000 モデル 50
項 目	B 1800	B 1801	B 1802	B 1803	B 1804
ステーション接続台数	1 台	8 台	1 台	4 台	8 台
メ モ リ 容 量	384〜896 K バイト				
ブ ロ セ ッ サ	16 ビット μ プロセッサ				
基 本 ソ フ ト ウ ェ ア	拡張日本語 MP/M-86™ システム 1		拡張日本語 MP/M-86™ システム 2		
固 定 デ ィ ス ク 装 置	—	—	15 M バイト×1 台 増設 1 台可能		38 M バイト×1 台 増設 1 台可
フレキシブルディスク装置 (基 本) (増 設)	5 1/4 インチ 1 M バイト×2 台		5 1/4 インチ 1 M バイト×1 台		
	5 1/4 インチ 1 M バイト×2 台 又は 8 インチ 1 M バイト×2 台		5 1/4 インチ 1 M バイト×3 台 又は 8 インチ 1 M バイト×2 台		5 1/4 インチ 1 M バイト×2 台 又は 8 インチ 1 M バイト×2 台
カートリッジ磁気テープ	—	—	—	—	20 M バイト
通 信 回 線	1 回 線			最大 2 回線	
そ の 他	RAS 機能 (パリティチェック, ECC チェック, バスタイムアウト, パワーフェイリュア, BAT) リアルタイム機構 (バッテリバックアップ), メロディアラーム機構				
消 費 電 力 (VA)	500		900		
電 源 条 件	100 V AC \pm 10 %, 50/60 Hz				
外 形 寸 法 (幅×奥×高) (mm)	150×480×425		300×480×425		450×700×700
重 量 (kg)	24		35		70

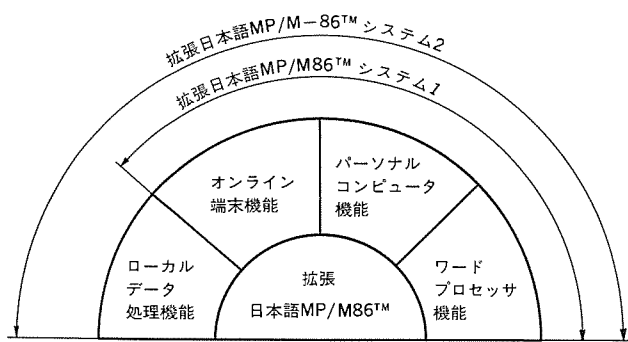


図 4. オフィスターミナルの基本ソフトウェア構成

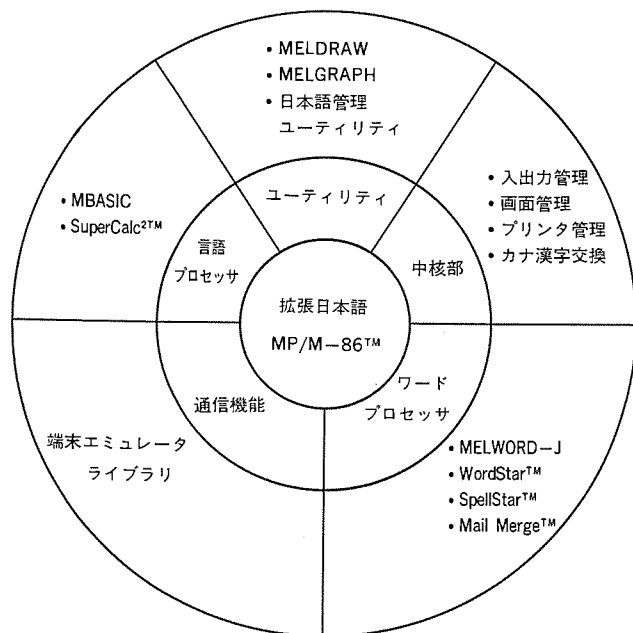


図 5. 拡張日本語 MP/M 86™ システム 1 の機能

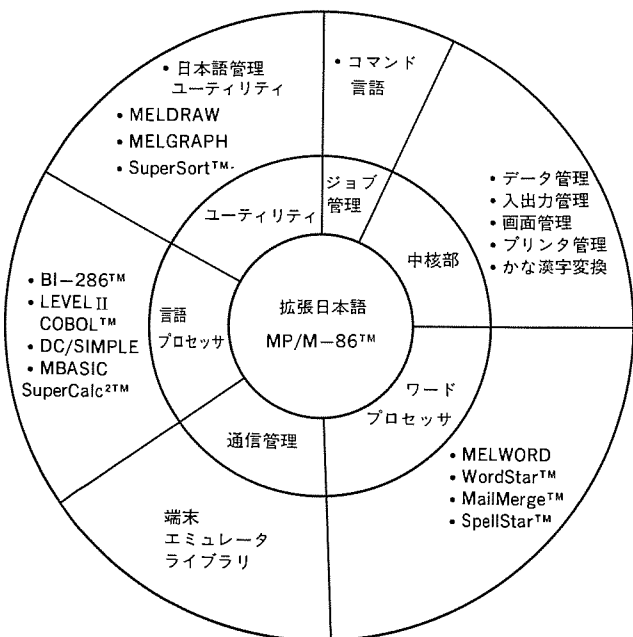


図 6. 拡張日本語 MP/M 86™ システム 2 の機能

フレキシブルディスクベースで使用される基本ソフトウェアであり、モデル 10 S/10 をサポートする。図 5. に システム 1 の機能を示す。

(2) システム 2

固定ディスクベースで使用される基本ソフトウェアであり、モデル 30 S/30/50 をサポートする。図 6. に システム 2 の機能を示す。

次に拡張日本語 MP/M-86™ の特長について述べる。

4. 1 マルチプログラミング制御機能

オフィスターミナルのように端末機能をもつシステムの基本ソフトウェアとしては、マルチプログラミングができることが必ず(須)である。254 レベルのタスクプライオリティ制御やタイムクォンタム制御などミニコン並みの実時間制御機能をもっている。

4. 2 日本語処理機能

パーソナルコンピュータ《MULTI 16》で扱う PC 漢字コードのほか、オフィスコンピュータやホストコンピュータで扱う MELCOM 標準漢字コードをディスプレイ装置、日本語プリンタ装置で扱うことができる。日本語入力方法も従来の区点入力、ヘキサ入力方法のほかに、かな漢字変換(ローマ字漢字変換を含む)入力機能をサポートしている。JIS 第 1 水準までの日本語は、ROM 化されているが、それ以外の日本語は外字扱いとなる。外字領域はディスプレイ装置、日本語プリンタ装置とも 128 種分ある。外字制御方法として、プリロード方式、オンデマンド方式の両方がサポートされている。

4. 3 ファイル管理機能

固定ディスクファイルをサポートするために、2 レベルの階層構造のファイルディレクトリ方式を実現している。この方式は、大容量ディスクファイルを効率よくアクセスするために必須なものである。また、高級言語のために次のファイル編成をサポートするデータ管理機能(システム 2 のみ)をもっている。

- (1) 順ファイル
- (2) ライン順ファイル(ソースファイル)
- (3) 相対ファイル
- (4) 多面索引ファイル(B-tree 方式)

データ管理機能を使用すると、プリンタへの出力や通信回線へのアクセスを順ファイルとして扱うことができる。また、論理ファイル名を使用して、物理ファイル名をプログラム実行時に ASSIGN コマンドで指定することも可能である。

4. 4 画面管理機能

オフィスコンピュータなどでは、一般的である適用業務プログラムのアクセスできる画面を、コンソール画面と業務画面に分ける手法を採用している。また、同時に二つのプログラム(ジョブ)が同時に独立して画面をアクセスできる。このような画面・ジョブの切替は、キーボードからワンタッチで行える。

4. 5 ジョブ管理機能(システム 2 のみ)

フォアグラウンド/バックグラウンドのジョブ管理を行う。簡単で使いやすいジョブコマンド、コマンドのカタログ機能、メニュー機能、高水準なコマンド言語(CL)機能などをサポートする。

4. 6 ローカルエリアネットワーク(LAN)管理機能

マルチステーション構成(モデル 30, 50)の場合、各ワークステーションで実行する適用業務プログラムが、ローカルなディスクファイルやプリンタをアクセスするのと同じ方法で、基本部のディスクファイル、プリンタステーション、他のワークステーションの日本語プリンタなどをアクセスする機能を提供する。また、ワークステーションがローカルにディスク装置をもたない場合、基本部から拡張日本語 MP/M-86™ をダウンロードする機能もサポートしている。

4. 7 オンライン端末機能

オフィスターミナルを《MELCOM》コンピュータや他社コンピュータのホストコ

コンピュータ用オンライン端末として利用するため、次のような端末エミュレータを用意している。

- ・M 4374 日本語エミュレータ
- ・DDX サポートソフトウェア
- ・その他

5. 言語プロセッサ

5.1 概要

三菱 オフィスターミナル 5000 シリーズには、LEVEL II COBOL™, DD/SIMPLE, BI-286™, MBASC の言語プロセッサが備えられている。LEVEL II COBOL, DD/SIMPLE, BI-286 は、基本ソフトウェアとして用意されているデータ管理機能を共通に使用しており、モデル 30 S/30/50 で使用可能である。また、MBASIC は全モデルで使用可能である。

5.2 LEVEL II COBOL

LEVEL II COBOL は、ANSI 規格 COBOL 標準 X 3.23 (1974) に準拠した言語機能を備えている。表 2. に ANSI 規格との比較を示す。LEVEL II COBOL の特長は次のとおりである。

- (1) 画面入出力、日本語、ファイルシェア/レコードロックなどの機能が拡張されている。
- (2) コンパイル後、直ちに実行可能であるので、プログラム開発時間を大幅に短縮することができる。

(3) 会話形デバッグが用意されており、デバッグが容易である。

(4) 画面定義プロセッサ (FORMS-2™) があり、画面設計、作成が容易である。FORMS-2 は次の機能をもっている。

- (a) 利用者が設計、作成した画面形式を COBOL の登録集として出力する。アプリケーションプログラムからは、COPY 命令を使用して参照する。
- (b) 画面形式をアプリケーションプログラムと結合することなくチェックするためのチェックプログラムを作成する。
- (c) 作成した画面形式のファイルを作成する。
- (d) 作成画面に基づく索引プログラムのソースプログラムを出力する。索引プログラムは、入力されたデータの格納、検索、更新及び削除の

表 2. COBOL の ANSI 規格との比較

機能単位	水準		
中核	水準1	水準2	
表操作	水準1	水準2	
順ファイル	水準1	水準2	
相対ファイル	水準0	水準1	水準2
索引ファイル	水準0	水準1	水準2
整列併合	水準0	水準1	水準2
報告書作成	水準0	水準1	
区分化	水準0	水準1	水準2
登録集	水準0	水準1	水準2
デバッグ	水準0	水準1	水準2
プログラム間連絡	水準0	水準1	水準2
通信	水準0	水準1	水準2

注 〇の部分の機能をもっている。

ために使用することができる。

図 7. に COBOL プログラムの開発の流れを示す。

5.3 DD/SIMPLE

DD/SIMPLE は、分散処理システムに必要な適用業務、すなわちデータエントリ処理、ビルディング処理、問合せ処理及びバッチ処理を行うための簡易言語であり、その特長は次のようである。

- (1) カラム形式で入力可能であるのでプログラム作成が容易である。
- (2) データエントリ処理の操作特性 (入力、更新、探索モードなど) をもち、画面入力時には項目単位の検査/編集/演算処理ができる。また、数値項目の少数点などの編集や明細の挿入/追加などが伝票形式と同様に入力、表示できる。図 8. に DD/SIMPLE プログラムの開発の流れを示す。図 9. は DD/SIMPLE の実行時の動きを示している。

5.4 BI-286

BI-286 は、事務処理用 BASIC であり事務処理に必要な次の機能を備えている。

- (1) 演算精度を損わないために、数値は内部的に BCD 形式で保持している。
- (2) ダイレクトファイル (索引ファイル)、インデックスファイル (相対ファイル) のアクセスが可能である。また、それらに対するファイル/レコードロック機能を備えている。

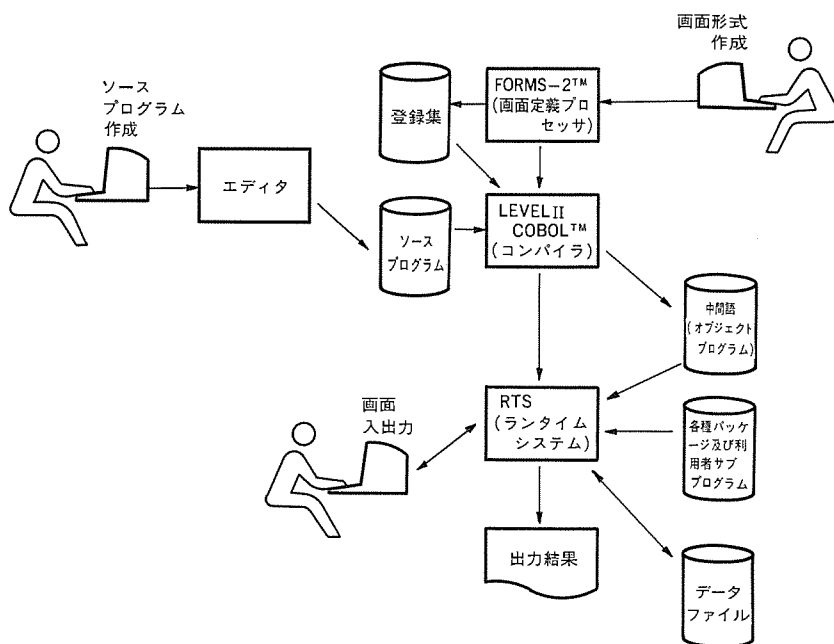


図 7. COBOL プログラムの例

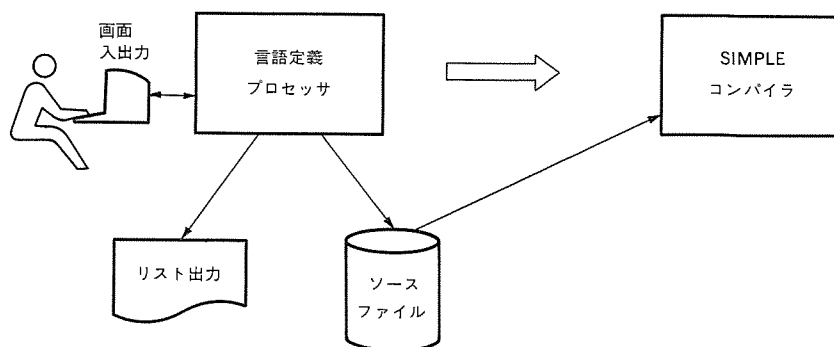


図 8. DD/SIMPLE プログラム開発の流れ

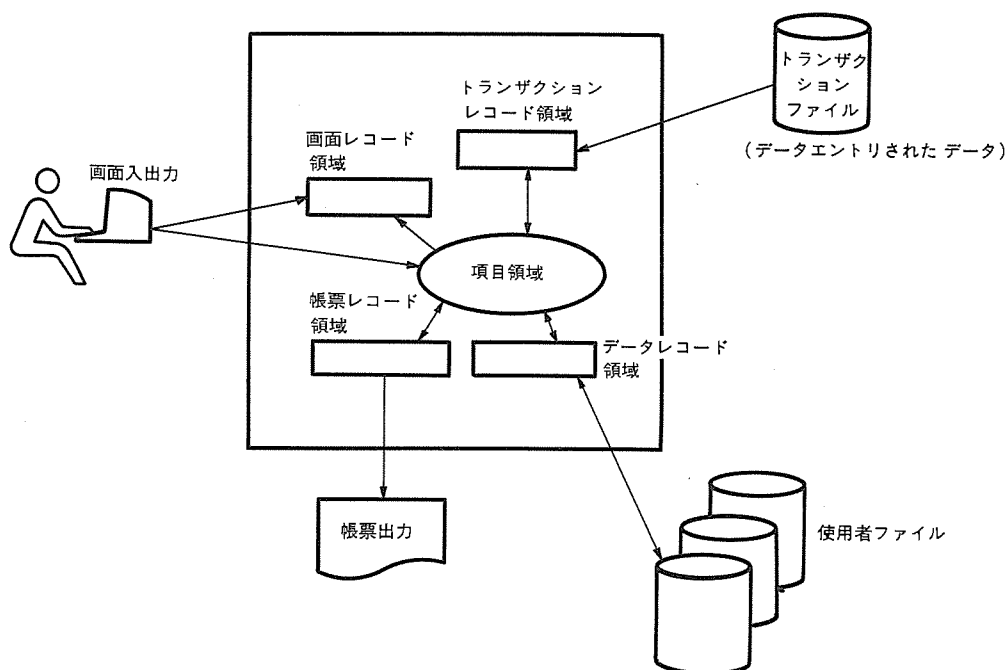


図 9. DD/SIMPLE の実行時概要

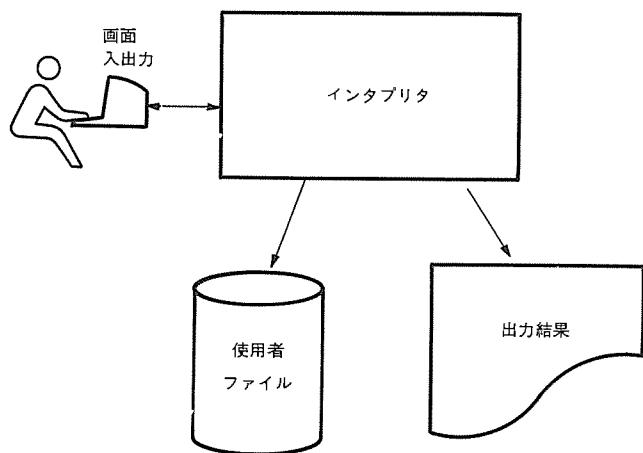


図 10. BI-286, M BASIC プログラムの開発の流れ

- (3) 画面入出力機能が拡張されている。
- (4) 日本語の入出力が可能である。
- (5) 複数ユーザーから同一サブルーチンを同時に使用できるパブリックプログラム機能を備えている。

5.5 MBASIC

MBASIC は、パーソナルコンピュータとして利用するときの BASIC であり、次の機能を備えている。

- (1) シーケンシャル/ランダムファイルのアクセスが可能である。
- (2) 機械語モジュールとのリンクが可能である。
- (3) 日本語処理が可能である。
- (4) 他装置とのコミュニケーションが可能である。
- (5) グラフィック機能が使用できる。
- (6) ミュージック機能が使用できる。

図 10. に BI-286, MBASIC プログラム開発の流れを示す。

6. OA パッケージソフトウェア

オフィスターミナル M 5000 シリーズのオフィスオートメーション(OA)システム構築

のために、用意されているソフトウェアを OA パッケージと呼び、次のものが用意されている。

- ・日本語ワードプロセッサ「MELWORD-J」
- ・英文ワードプロセッサ「Word Star™」
- ・会話形グラフ作成ユーティリティ「MELGRAPH」
- ・会話形図形作成ユーティリティ「MELDRAW」
- ・スプレッドシート「Super Calc™」
- ・ビジネスグラフィブラリ「BGL-C, -B」

図 11. に OA パッケージソフトウェアの体系を示す。

6.1 日本語ワードプロセッサ

かな漢字変換やローマ字漢字変

換機能及び充実した文書作成機能をもつ本格的な日本語ワードプロセッサである。数値データの計算処理はもちろんビジネスグラフデータや図形データを含んだ文書を作成することもできる。以下に機能をまとめて示す。

- (1) 編集機能……罫線、行組み、ブロック操作、スクロールなど
- (2) 数値データ処理機能……作表、計算及びソーティングなど
- (3) 印刷機能……差込み、袋とじ、縦書き、部分印刷など
- (4) その他……文書編集と印刷の同時並行処理、「MELGRAPH」、「MELDRAW」のグラフ及び図形データの文書中への差込み印刷など

図 12. に日本語ワードプロセッサで編集した文書の例を示す。

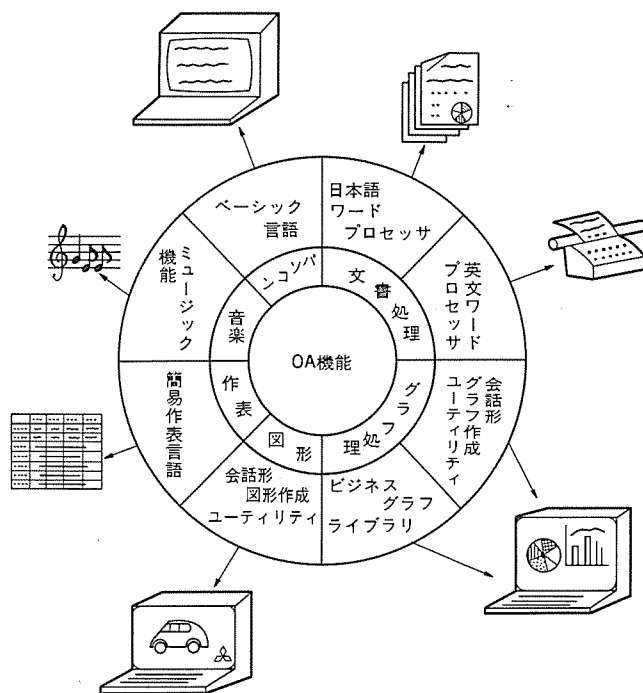


図 11. OA パッケージソフトウェアの体系

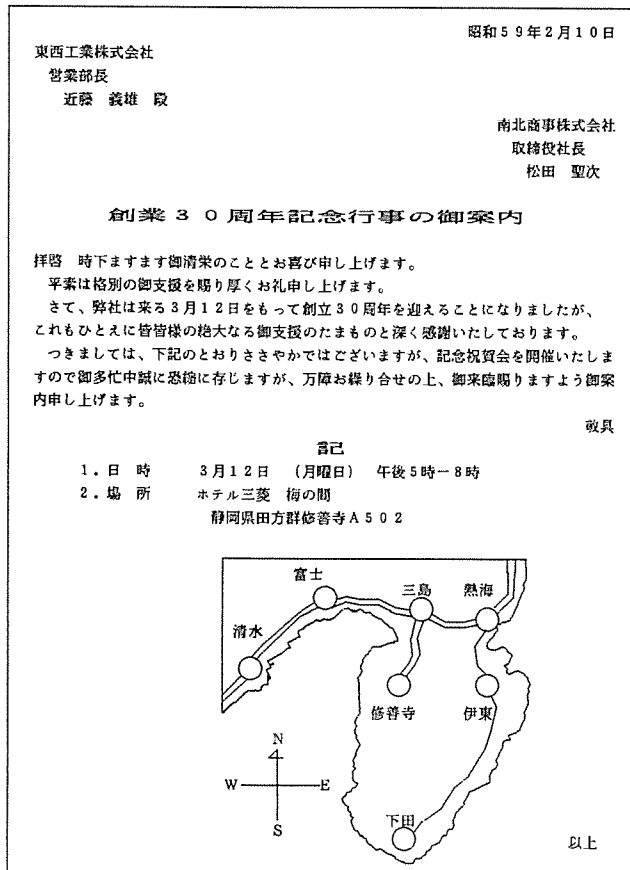


図 12. 日本語ワードプロセッサで編集した文書の例

6.2 英文ワードプロセッサ

欧米で最も一般的なワードプロセッサとして利用されている Word Star™ をサポートしている。また、Word Star™ で作成された文書を辞書と照合してスペリングチェックを行う Spell Star™、ファイルや入力データ（例えば住所録など）を文書中に挿入する Mail Merge™ を使用することができる。

6.3 会話形グラフ作成ユーティリティ

画面に表示されたコマンドを選択し、ビジネスグラフの数値データやタイトル、グラフの種類、軸の単位などのパラメータを入力するだけでビジネスグラフを作成することができ、次の機能をもっている。

(1) グラフの種類

使用目的に応じて、棒グラフ（縦／横）、積上げ棒グラフ（縦／横積上げ）、折れ線グラフ、円グラフ（円、内円付き円、部分切出し円）、内訳けグラフ及びレーダーチャートの6種類から選択することができる。また、作成した二つのグラフを同時に重ね合せ、又は分割表示することもできる。

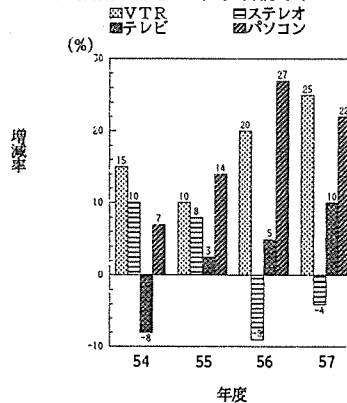
(2) その他

陰影パターン（16種）、線種（8種）、線幅（2種）、プロットマーカ（8種）、カラー（7種）を指定することができる。

6.4 会話形図形作成ユーティリティ

標準 JIS 配列キーボードから任意のサイズの円、円弧、四角形などの簡易図形やペン書き（フリーハンド）図形を作成することができる。また、漢字文字を入力することもでき、次の機能をもっている。

製品別売上高増減率



売上状況

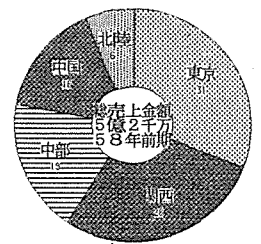


図 13. ビジネスグラフライブラリで作成したグラフの例

(1) 図形入力機能

直線、円、円弧、扇、ひし形、接線円などを指定された色、線種、線幅、大きさで描画する。

(2) 文字入力機能

指定された文字サイズ、回転で英数字、カナ、漢字文字を入力することができる。

(3) ペン書き

カーソルを移動しながら任意のフリーハンド図形を描画することができる。

(4) その他

図形データファイルの格納、再編集及び印刷の機能をもつ。

6.5 スプレッドシート

横63列、縦254行の格子状に区切られたワークシートに、日本語文字や数値データを入力し、表を作成することができる。このワークシートには、四則演算や標準組込み関数を用いて計算式を定義することができ、数値を入力又は変更すると自動的に再計算し、新しい結果が表示される。このワークシートの各要素は、コマンドにより移動、複写、挿入及び削除することができる。

6.6 ビジネスグラフライブラリ

アプリケーションプログラムから CALL 命令で呼び出すことによって、ビジネスグラフや任意の図形を作成することができるライブラリであり、LEVEL II COBOL 用の BGL-C と MBASIC 用の BGL-B の2種類が用意されている。図 13. にビジネスグラフライブラリで作成したグラフの例を示す。

7. む す び

以上三菱オフィスミナル M 500 シリーズの概要について紹介したが、今後この種の端末システムの需要はますます増加するものと考えられる。

残された課題として、各種端末エミュレータの充実・言語プロセッサの生産性向上・本格的 LAN (ローカルエリアネットワーク) への対応・イメージ処理への対応などがあげられる。最後に、このシステムの開発にあたって、多大な御指導と御支援をいただいた関係者各位に深く感謝の意を表する次第である。

3.5インチマイクロフレキシブルディスク装置

今村 恒 浩*・佐々木 勝**

1. ま え が き

フレキシブルディスクドライブ (以下、FDD) は、1972 年 9 月に IBM 社から発表され (ディスク 直径 8 インチ), その後, OA 機器, パーソナルコンピュータの発展に伴い, 飛躍的に需要が伸びてきている。技術的には, 小形化, 高密度化の歴史をたどり, 1976 年 9 月に米国のシュガート社が 5.25 インチ 形 FDD (以下, ミニ FDD) を発表し, 手軽なコンパクトさと低価格が非常な魅力となって市場に受け入れられ, 両面化, 倍密度化など, 性能がどんどん向上し, 5 インチ, 8 インチを含めた, 全 FDD は 560 万台 (1982 年世界) の出荷量に達し, 年率 40 % の成長が見込まれるまでに至っている。

最近になって 3~4 インチサイズのフレキシブルディスクを用いた小形 FDD が相次いで発表されたが, まず 1980 年末に国内で 3.5 インチ, 1981 年末に同じく国内メーカー 3 社から 3 インチ, 1982 年に米国の媒体メーカーグループから 3.25 インチ, 1983 年初めに米国 IBM 社が 4 インチを発表した。

当社はこれら 3~4 インチの中で, ミニ FDD 系列製品と同じ記憶容量が比較的实现しやすく (ミニ FDD とソフトウェアとフォーマット互換性が保てる), 硬いカートリッジに入っていて, 取扱が容易な, 3.5 インチサイズのフレキシブルディスク (通称マイクロフレキシブルディスク) を使用するマイクロ FDD (MF 351 形片面仕様) を開発した。その仕様, 構成, 設計思想を以下に述べる。

2. 市 場

図 1, 図 2. に示すとおり, FDD の生産台数は, テレビ, VTR に匹敵する勢いであり, それは今までむしろ専門的に使われていた情報関連機器が一般に普及しだしている傾向を示している。3~4 インチマイクロ FDD 開発にあたっては以上の点を考慮して, 小形で取扱が容易な装置を設計する必要がある。

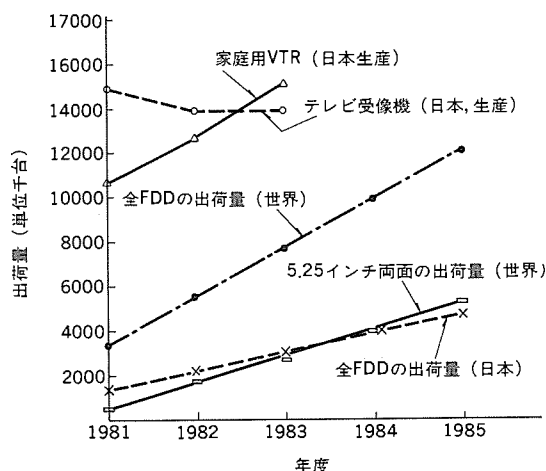


図 1. フレキシブルディスク装置の出荷予想 (全機種合計)

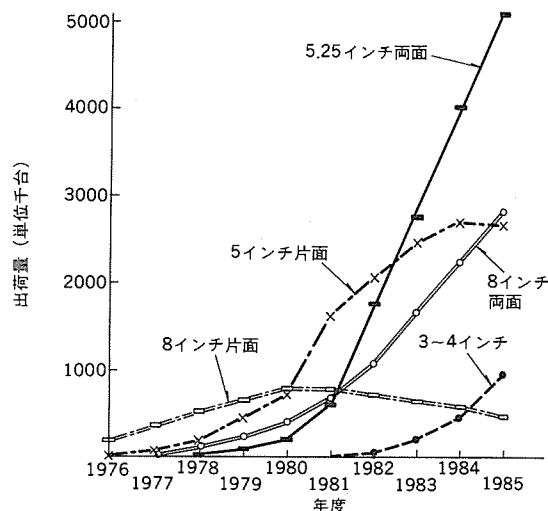


図 2. フレキシブルディスク装置の出荷予想 (機種別)
(出典 1982 DISK/TREND REPORT)

3. 構成と仕様

FDD は, ランダムアクセス機能と電源が切れてもデータが消滅しない, いわゆる不揮発性記憶装置という本質的な特徴のほかに, 媒体が小形で交換可能, 取扱の容易さ, 低価格を柱として需要を伸ばしてきた。

今回開発したマイクロ FDD (図 3. に示す) は, ミニ FDD で集積されたソフトウェアを生かし同じコントローラが使えるように, フォーマット, インタフェース信号をミニ FDD と同じにした。また取扱やすくするため小形にし, 大きさを 102 mm (幅) × 41 mm (高さ) × 145 mm (奥行) とし, 重量 700 g (ミニ FDD 薄形の 1/2) にした。図 4. に従来機種との比較 (FDD の大きさの歴史) を示す。

3.1 構成

マイクロ FDD は従来の FDD と同じく, ホストコンピュータからディスクコントローラを介して命令を受け, 指定されたディスク上に情報の書込み,

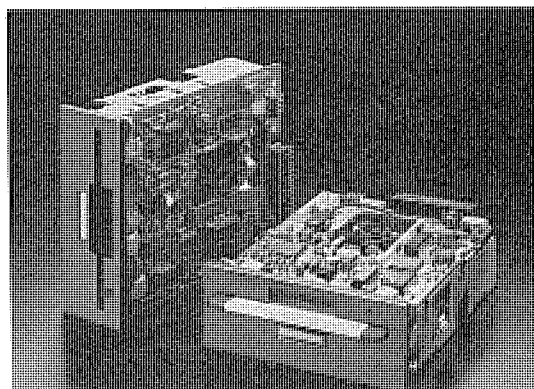


図 3. マイクロ FDD (MF 351 形) の外観

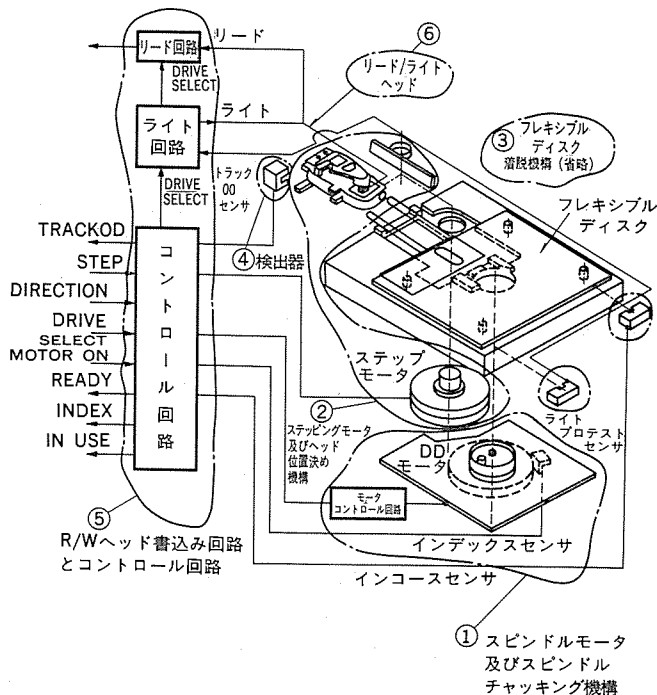


図 4. マイクロ FDD の構成ブロック図

表 1. 3~4 インチ FDD の仕様比較

	3 インチ*	3.25 インチ	3.5 インチ*	4 インチ
アンフォーマット容量/面 (K バイト)	250	500	500	358
記録方式	MFM	MFM	MFM	FM
記録面	片面/両面	片面	片面/両面	片面
最大ビット密度 (BPI)	8,946	9,250	8,717	6,865
トラック密度 (TPI)	100	140	135	68
トラック/面	40	80	70/80	46
回転数	300	300	300/600	262~415
データ転送速度 (Kb/s)	250	250	250/500	333
カートリッジ寸法 (mm ³)	80×100×5	83×87×2.15	90×94×3.4	120×120×5
シャッタ有無	あり	なし	あり	なし
センターハブ	プラスチック	金属	金属	なし

注 * 3 インチ, 3.5 インチの数字は MFM 倍密度記録の場合のもの。

表 2. マイクロ FDD (MF 351 形) の仕様

形 名	MF 351
アンフォーマット容量 (K バイト)	500
片面 / 両面	片面
最大ビット記録密度 (BPI)	8,187
トラック密度 (TPI)	135
トラック / 面	80
データ転送速度 (Kb/s)	250
回転数 (rpm)	300
平均回転待ち時間 (ms)	100
トラック間アクセス時間 (ms)	3
平均アクセス時間 (ms)	79
セトリング時間 (ms)	15
モータ起動時間 (s)	0.5
外形寸法 (mm)	41×102×132
重量 (g)	700
電源 +12 V (A)	0.4
+5 V (A)	0.6

読出しを行うもので、その構成は図 4. に示すようになっていて、次の要素から構成される。

- (1) スピンドルモータ及びスピンドルチャッキング機構
- (2) ステップモータ及びヘッド位置決め機構
- (3) フレキシブルディスク着脱機構
- (4) 各種検出器
- (5) R/W ヘッド書込み、読出し回路とコントロール回路
- (6) R/W ヘッド

3.2 仕様

3~4 インチ系 FDD の仕様の比較を表 1. に、MF 351 形の仕様を表 2. に示す。

4. フレキシブルディスク (媒体)

3~4 インチ系の特長はフレキシブル、次のとおりである (図 5. 3.5 インチフレキシブルディスク形状参照)。

- (1) ケースが硬いカートリッジになっている (3.25 インチは従来と同じソフトなケース)。
- (2) R/W ヘッド用ウインドーはシャッタ付き。
- (3) 装置への装着部分は、金属ハブとし (図 6.) 装着時のガタをなくして位置ずれを少なくした。

装置はこの特長を十分に生かした設計にする必要がある。以下に今回開発した装置に使用する 3.5 インチのフレキシブルディスクの仕様について述べる。

4.1 磁気ディスクの形状、材料

磁気ディスクは外径 80 mm で、厚さは従来の 5 インチ、8 インチと同じ約 80 μ m である。金属ハブの形状機能については後述する。ハードケースとの関係は図 5. に示す。ベースは従来と同様ポリエチレンテレフタレート

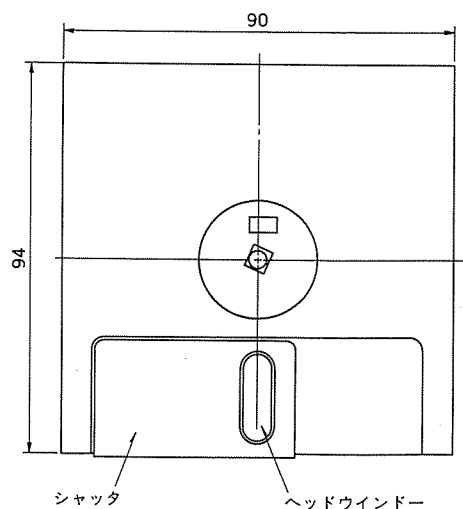
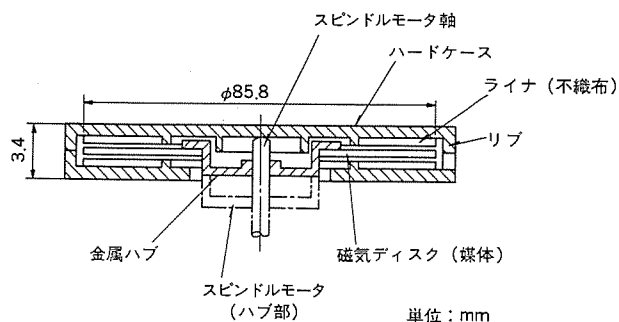


図 5. フレキシブルディスクの形状

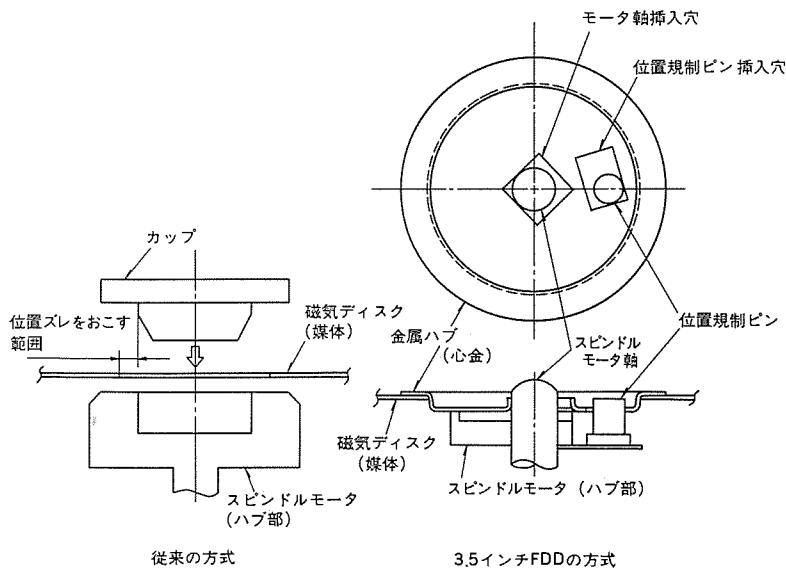


図 6. チャッキング方式の違い

(PET)を用い、磁性層は高密度記録のため高抗磁力材料の $\text{Co}-\gamma\text{-Fe}_2\text{O}_3$ を用い、抗磁力(H_c)は500エルステッド程度である。また磁性層厚さは約 $1.9\mu\text{m}$ で高線記録密度を可能にしている。

4.2 ハードケース、シャッター

従来のフレキシブルディスクのような塩化ビニルジャケットと異なり、プラスチック製のハードケースに収納されている。また従来のフレキシブルと同様に、シートに付着したゴミなどを取るために不織布を用いているが、最大負荷トルクは60gr-cm前後と、従来の5.25インチミニフレキシブルディスクの1/5程度となっている。

R/Wヘッドが入るヘッドウインドーは、金属のシャッターが設けられていて、装置に装着すると自動的に開き、装置から取り出すと自動的に閉じるようになっていて、使用時以外は閉じてゴミが入らないと同時に指が触れないようになっている。

4.3 チャッキング機構と位置決め精度

トラック数は80本、トラック密度は5インチフレキシブルディスクに比べ1.4倍の135TPI(Tracks per Inch)である。トラック番号の00(最外周)の位置は半径で39.5mm、最内周トラック位置は24.7mmである。なお、両面ヘッドのドライブの場合、反対側の面のヘッドにおける最内周トラック位置は更にトラック8本分(1.5mm)内側に位置している。

この装置はミニFDDとインタフェース信号がコンパクトに使えることが特長であるが、ソフトウェアが使えることももちろんであって、そのためには記録フォーマットが同じでなければならない。これを実現するためには、トラックの記憶容量とデータの転送速度がミニFDDと同じであることが必要である。ミニFDDの場合、トラック当りの記憶容量は、6.25Kバイト(MFM記録方式)で、データ転送速度は250Kb/s(キロビット/秒)である。

この仕様に合わせて最大記憶密度は、トラック番号79(最内周トラック)において、8,187ビット/インチ(MFM記録方式)となる。回転数は300rpmである。3.5インチ形では、新しくシートの中心に心金(金属ハブと称す)を入れ、心金の中央の穴が四角形で、隣接する二辺が必ずモータ軸と接する構造により、チャッキング時の位置ズレが無視でき、135TPIと言う高トラック密度で十分な互換性が得られ

るようになっている。

5. 装 置

大きさ(体積)の推移を図7.に示したが、この中で最小の大きさを目指とした。装置外形幅は業界標準に合わせ、102mmとした(フレキシブルディスク外形幅90mm)。高さは、5.25インチFDDの薄形タイプと同じ41mmとした。奥行は、業界標準に合わせ、150mmとしたが、将来的には更に小形化する方向で検討したい。

以上実現のための技術的課題は次のとおりである。

- (1) スピンドルモータ及びステップモータの小形、薄形化
- (2) フレキシブルディスク着脱機構
- (3) 回路基板の小形化
- (4) 発熱や磁界発生などのノイズ低減

5.1 偏平形スピンドルモータ

ミニFDDで採用され始めた偏平形ブラシレスダイレクトドライブモータを本機でも採用した。その理由はベルトレスとなり、装置内のスペースが有効に活用でき、保守が不要で長寿命であり、装置の組立調整が容易なことである。更に今回薄形化のために、ステータの鉄板を延長し、その上にフレキシブルプリント板をラミネートしてモータコントロール回路を搭載し、従来のプリント回路基板を排除し、かつ電子部品をチップ化して厚みを9mmとした。

5.2 ステップモータ

ステップモータ小形化が、FDD小形化の最大ポイントであって、従来のものに比べて約1/3の 17cm^3 ($D=\phi 46\text{mm}$, $t=10\text{mm}$)にした。

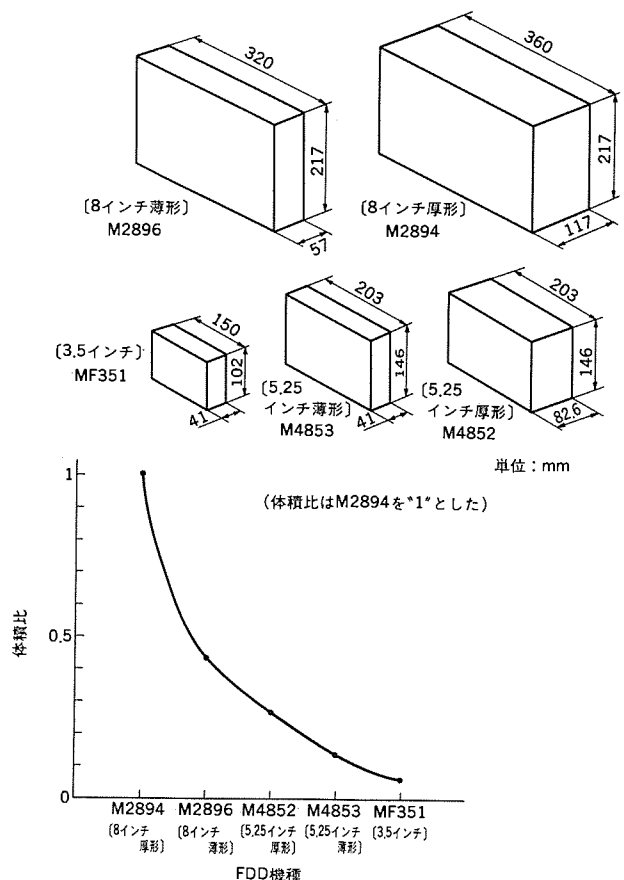


図 7. FDDの大きさ(体積)の歴史(当社比)

これはキャリッジなどの負荷を従来の1/3にしたこと、ステップモータの歯部の形状を改良して磁気効率をあげたことによる。

5.3 フレキシブルディスク着脱機構

操作性の良さが最も要求される場所である。着脱には、より信頼度の高い専用のポケット状のホルダを設け、フレキシブルディスクの挿入と共にホルダがスピンドルの方向に移動して、ディスクをスピンドルの上に装着させる。排出はエジェクタボタンを押すことにより、ディスクホルダを押し上げた後、ディスクが飛び出すようにして、ワンタッチオペレーションを可能にしている。

5.4 回路基板

従来品は、リードライト回路及びコントロール回路を基板1枚に納め、これをFDD上部全体に装着していたが、装置が小さくなり、部品実装が不可能になったことから回路の大幅なLSI化を図り、回路基板の表面積を60%以下に減らすことができた。また従来より一周り小さい電子部品の採用により、厚みも8mmから6.5mmとした。

5.5 発熱、磁界発生などの影響

従来、リード、ライト時以外は、ソレノイドにより、媒体をヘッドに押しつけているパッドを持ち上げていた。これは媒体とヘッドの回転接触時間を減らし寿命を伸ばすためである。近年スピンドルモータの性能向上により、スタート、ストップ時間が速くなり、リード、ライト時以外はモータを止めて摩擦時間を少なくする方法を採用しても、それ程処理時間に影響を与えなくなったので、ソレノイドを除去した。これで主な発熱源の一つを除いている。ステップモータも動作時のみ12Vを印加し、静止時は5Vに下げる方式とした。またステップモータの周りをベースの補強リブで囲い放熱効果をもたせた。ICもデジタル回路部分をCMOS-LSI化し消費電力を下げた。

磁気ヘッドが磁界の影響を受けると、波形非対称性によるピークシフトや減磁が発生し、エラー率が増加する。発熱対策によるソレノイド除去、ステップモータ保持電圧の5V化は、発生磁界減少にも役立ち、高信頼性磁気記録に大きく寄与している。

6. 高密度化

6.1 トラック密度

FDDのエラー率が増加するのは、異なる環境条件で、異なる装置によって書き込まれたデータを読むときである。高密度化の課題は、この互換によるエラー率の増加を如何に防ぐかであり、その方法はオフトラックによってエラー率が増加しないようにすること、メカニカルなオフトラックを最小限にすることである。

6.1.1 オフトラックマージン

オフトラックすると（書いたデータ上からヘッドが半径方向にずれて読むこと）エラー率が増加する。その主な原因はS/N低下によるビットシフトの増加である。オフトラック許容幅はミニFDDが約100μmに対して3.5インチ形では約70μm（実測）である。これ以上ずれるとビットエラー率 10^{-9} が保証できなくなる。互換性を得るためには各装置は、最悪ケースを考慮して±35μm以下にヘッドの位置ずれを抑えなければならない。

6.1.2 オフトラックの要因と対策

3.5インチと小形でありながら5.25インチと同じ容量にできたのは、半径方向の記録密度（トラック密度）をあげるために、ヘッドと媒体の相対的な位置ずれ（オフトラック）を小さくできたからである。

FDDはヘッドのトラックアクセスに、ステッピングモータを用いるオープン

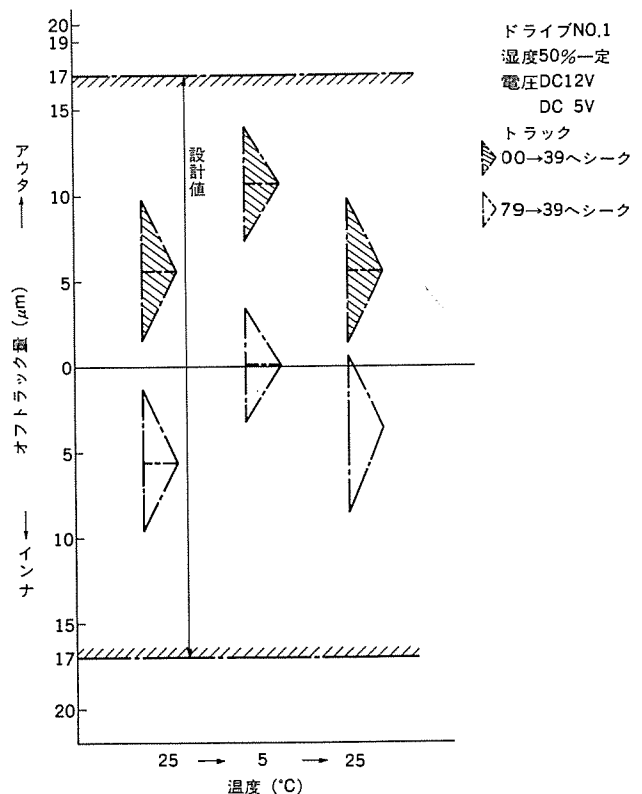


図8. MF 351 温度 オフトラック 測定結果（バックラッシュ含む、湿度、電圧変動は除く）

ループ制御方式が一般的であるため、ステップモータの角度誤差などが直接精度にきいてくる。また媒体は可とう性のあるPETの薄いフィルムをベースとして使うため、温度、湿度による膨脹、収縮の影響も大きい、などオフトラックする要因は多い。その主なものをあげると、

- (1) 温度変化、(2) 湿度風化、(3) トラックピッチ精度、(4) 調整誤差、(5) ヘッド上下差（両面形ヘッドの場合）

(1) 温度変化によるオフトラック

ディスクは、厚さ約75μmのポリエステルベース材の両面に1~2μmの磁性層を塗布したものであるが、温度変化による膨脹収縮は、ベース材の熱膨脹係数によって決まる。今回使用の3.5インチのPETの熱膨脹係数は $\alpha_T = (17 \pm 8) \times 10^{-6} \text{ mm/mm/}^\circ\text{C}$ である。装置としての温度オフトラックの要因は媒体、ベース、キャリッジなどがあり、これらの材質と構成の組合せを適当に選び、零に近くすることが望ましい。MF 351は図8.に示す実験結果のように5μm/20°Cとなっている。

(2) 湿度変化によるオフトラック

PETの湿度膨脹係数の実情は $\alpha_H = (10 \sim 11) \times 10^{-6} \text{ mm/mm/\% RH}$ となっている。しかしながら動作中のFDDに実装してオフトラック量を測定すると、その値はほぼ $\alpha_H = (6 \pm 3) \times 10^{-6} \text{ mm/mm/\% RH}$ となっている。これはディスク面の湿度が、装置内温度上昇や、ディスクとライナとの摩擦熱の影響で実効湿度が変化しているためと考えられる。これに反してFDD（装置）は、構成材料の寸法が湿度によって変化しないので、湿度による膨脹に対して補償は行っていない。しかし出荷時のヘッドの位置調整を一定湿度で行うなどの工夫で影響を最小限にとどめている。

(3) トラックピッチ精度

高速アクセス化とトラックピッチの精度向上にメタルバンド方式を採用し

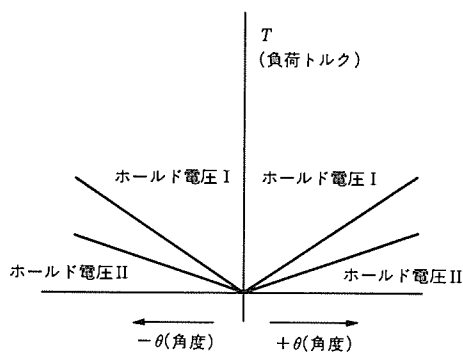


図 9. ステップモータ θ - T 特性

た。誤差要因は、①キャプスタン 直径誤差、②キャプスタン 軸振れ、③バンドの厚みのばらつき、④スピンドルモータの軸振れ、⑤ステップモータ 角度誤差である。この中でもっとも大きいのが⑥のステップモータ 角度誤差である。角度誤差のうち、取扱に注意を要するのが θ - T 特性(図 9.) である。内周側から進んだときと、外周側から進んだときでは摩擦負荷が反対に働き、位置のズレがそれぞれ相反した形で表れる。出荷調整のときに両方向から送って、その中心にちょうど位置合せするなどの工夫をしなければならない。またステップモータの各相間の角度精度も小さくする必要があり、通常このばらつきは 3 % 以下にするが、歩留りとコストとのからみからみれば、かなり厳しい値である。

以上を総合したオフトラック(除く湿度、電圧変化)は、図 8. の実験結果に示すように設計値を満足している。

6.2 線記録密度(磁気ヘッド)

線記録密度の向上は、媒体のほか磁気ヘッドの特性に依存する。従来の 5.25 インチ FDD に比べて、最内周トラック半径が小さく、かつトラック幅が狭くなった分、再生電圧が小さくなり S/N が悪化する。出力電圧 E は次式によって示される。

$$E \propto v \cdot N \cdot T_w \cdot \frac{B_r \cdot \delta}{a+d} \cdot \eta \cdot L$$

v : 媒体スピード
 N : コイル 巻数
 T_w : トラック幅
 B_r : 媒体残留磁束密度
 δ : 媒体磁気膜厚
 a : 残留磁化転位幅
 d : ヘッド 浮上すき間
 η : ヘッド 再生効率
 L : ヘッド 損失

既開発の 1.6 M バイト 5.25 FDD と比較すると、媒体特性を同一と

表 3. リードライト 特性

		設 計 目 標	結 果 \bar{x} $n=5$
出 力 (mV)	$f_{TR}=00$	2.0 以上	3.0
	$2f_{TR}=79$	0.7 以上	1.0
分 解 能 (%)		65 以上	74.2
オ ー バ ラ イ ト (dB)		30 以上	32.5 ($n=3$)
タイミグマージン (μs)		0.8 以上	1.08

みて、ヘッドギャップ及び巻線仕様を変えない場合は、出力が 40 % まで低下する。線記録密度は 1.6 M バイト FDD の 9,646 BPI と比べて 8,187 BPI とわずかに低いので、ギャップ長を更に小さくする必要はない。したがってヘッドコイルの共振周波数低下の許容範囲内で、コイル巻数補正を行うなどにより、出力電圧 1 mVp-p 程度が確保でき十分な S/N 比を得ている。その結果、表 3. に示すように十分目標の特性を得ている。

7. む す び

3~4 インチ FDD は、

- (1) 専用 IC 小形モータなどを用いた場合より一層の小形化
- (2) 電池駆動可能な低消費電力化
- (3) プラスチック部品の多用などによる低価格化
- (4) 両面化により 1.0 M バイトの実現
- (5) 更に 1.6~2.0 M バイトも 3.5 インチサイズで遠からず実現可能と考えられている。

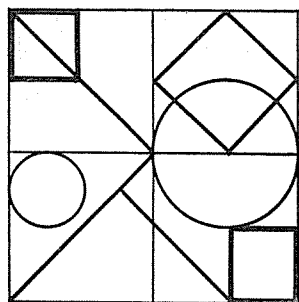
など、ますます多岐にわたって製品化され、大量に使われていくものと考えられる。また垂直記録方式としても生産性に優れ、耐久性の良い垂直記録媒体が実現すれば、更に使いやすい大容量メモリが可能となろう。

参 考 文 献

- (1) 1982 DISK TREND REPORT
- (2) 公開特許公報、昭 57-36463「記録再生装置」ソニー
- (3) A Proposal for an American National Standard for an Unformatted 3.5 Inch.

特許と新案 有償開放

有償開放についてのお問合せ先 三菱電機株式会社 特許部 TEL (03) 218-2136



ディスプレイ装置 (特許 第1166190号)

発明者 小田 勇介・渡辺 治

従来のライトペンを用いたディスプレイ装置は、選択された項目の表示位置をレジスタにセットする方式をとっていたため、1項目選択するごとに計算機に対してデータを送信せねばならず、オペレータに負担がかかり、計算機との迅速かつ正確な対話が困難であるという欠点があった。

この発明は、CRT上にライトペン検出可能文字をパターンで表示しておき、このパターンをライトペンで指示した場合に、ライトペン検出可能文字をライトペン検出文字に置き換えるよう構成することにより、一つの表示画面上でライトペンによる複数個の項目選択を可能にし、かつ選択した項目の確認ができ、正確なデータを一括して計算機に送信できるように改善したものである。

図1. はこの発明の一実施例を示す基本ブロック図で、計算機とのデータの受け渡しやキーボード(8)及びライトペン(9)からの信号の処理を行う処理回路(1)、メモリ制御回路(2)、メモリ(3)、キャラクタジェネレータ(4)、ビデオ信号発生回路(5)、タイミング信号発生回路(6)、表示装置(7)を備えたものにおいて、メモリ(3)から得たライトペン検出可能文字コード又は同検出文字コードからそれぞれの表示パターン信号を発生する回路(29)を設けると共に、上記処理回路(1)を、ライトペン(9)の検出信号によりライトペン検出可能文字から同検出文字へのメモリ(3)の書換え、又は逆の書換えを指示し、計算機へのデータ送信の場合にライトペン検出文字のポジションを送信する働きを行うよう構成したことを特長としている。

図2. はこの発明による項目選択画面例を示し、斜線の四角がライトペン検出可能文字パターンで、空白の四角が同検出文字パターンである。図では「1 ネンレイ」「2 セイベツ」「3 ショクギョウ」までがそれぞれ

一つずつ選択され、「4 ジュウシヨ」以降が選択されていない状態を示している。

このように、ライトペンで指定された表示位置をレジスタ類に記憶するのではなく、メモリのデータを書き換えることにより記憶するから、一つの表示画面上で多項目の選択が可能となり、計算機との対話が円滑になる。また、選択した項目と選択していない項目との表示がはっきりと区別されるから、選択ミスのまま計算機に送ってしまうことを避ける効果がある。

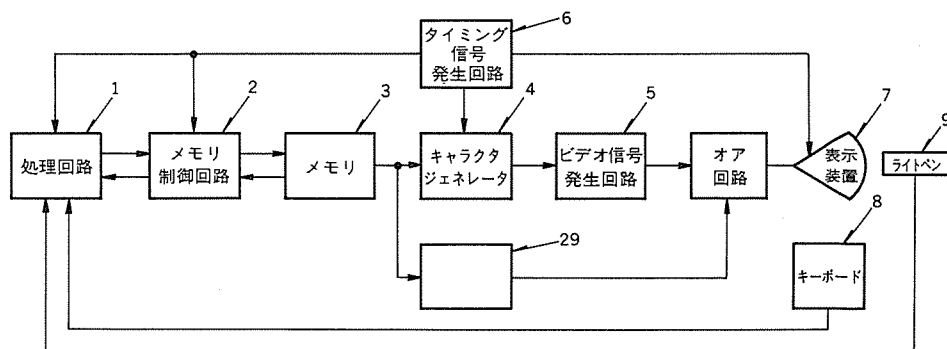


図1

ナマエ 「*****」

1 ネンレイ	■ ~20	□ ~30	■ ~40	■
2 セイベツ	□ M	■ W		
3 ショクギョウ	■ コウムイ	□ ギシュ	■ ジム	■ キョウシ
4 ジュウシヨ	■ ホッカイ	■ トウホク	■ カントウ	■ チュウフ
	■ カンサイ	■ チュコク	■ シコク	■ キュウシ
5 シツモン 1	■ 1	■ 2	■ 3	■ 4
6 シツモン 2	■ 1	■ 2	■ 3	■ 4
7 シツモン 3	■ 1	■ 2	■ 3	■ 4

図2

バス占有要求調停方式 (特許 第1161036号)

発明者 保刈 明彦

この発明は、同一I/Oバスに接続された複数の装置間のデータ転送を行う際に、各々の装置から発生するバス占有要求を調停する方式に関するものである。

従来の方法は、より優先順位の高いバス占有要求が来ると、既にバスが占有されていても、占有許可を取り消し優先順位の高い方に即座に許可を与えるといった優先順の調停を行うものが考えられて

特許と新案 有償開放

有償開放についてのお問合せ先 三菱電機株式会社 特許部 TEL (03) 218-2136

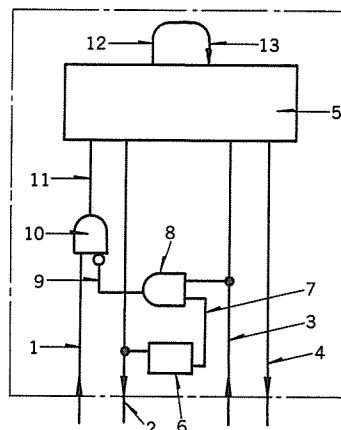
いた。しかし現実には、たとえ優先順位が低くとも一旦バスを占有したら、ある一定時間は確保する必要がある、それ以降は途中で打ち切られても構わないといったものも取扱う必要がある。

この発明は、簡単な回路で、これができるように改善したものである。

図はこの発明の実施例回路図であり、(1)は低優先順位バス占有要求信号、(2)は同信号(1)に対する許可信号、(3)及び(4)は高優先順位バス占有要求信号及び同許可信号、(5)は従来通りの先着順バス占有要求調停回路、(6)はバス占有許可信号(2)により時間ゼロにリセットされるタイマで、所定の時間経過するとタイマ信号(7)を発生する。(8)は上記要求信号(3)及びタイマ信号(7)が共に発生している時に禁止信号(9)を発生するゲート回路、(10)はこの禁止信号(9)により要求信号(1)が先着順調停回路(5)に伝わるのを禁止するゲート、(11)は禁止されていない時にゲート(10)から調停回路に加わるバス占有要求信号、(12)は要求信号(3)又は(11)の少なくとも一方が発生していることを示す信号、(13)は調停回路(5)を働かせるためのイネーブル信号である。

上記構成においては、タイマ(6)の設定時間内に高優先順位バス占有要求(3)が来ても、既にバスが占有されているときにはその終了まで待たされる先着順の調停をし、タイマの設定時間経過後も引き続

きバスが占有されている場合には、タイマ信号(7)と高優先順位要求信号(3)とにより禁止信号(9)を発生して、前者のバス占有を打ち切り、後者のバス占有要求に対して占有を許すように動作する。したがって、優先順位が低くともいったんバスを占有すると、ある一定時間占有権を確保することが望まれるようなバス占有要求に対し、バス占有を保証する調停を行うことが出来る。



〈次号予定〉 三菱電機技報 Vol. 58 No. 7 光応用特集

特集論文

- 光通信技術の展望
- 通信用半導体光デバイスの展望
- 光応用計測技術の展望
- 通信用半導体光デバイス
- 光通信用光部品とその応用
- 光送受信器とその応用
- 4波波長多重通信システム
- 光ファイバセンサ——強度変調形——

●光ファイバセンサ——レーザ干渉形——

●レーザ式パイプ表面検査装置

●近赤外撮像暗視システム

普通論文

- 大容量タービン発電機の回転子制動回路
- トルクセンサ内蔵電動機
- 国際電信電話(株)向けDS-11形非同期端末パケット集線装置(CPAD)
- GDS-70モデルDS図面作成支援システム

三菱電機技報編集委員

委員長	馬場 準一	委員	山内 敦
副委員長	岸本 駿二	"	柳下 昌平
"	三浦 宏	"	櫻井 浩
委員	峯松 雅登	"	徳山 長
"	翠川 祐	"	柴山 恭一
"	佐藤 文彦	"	酒井 靖夫
"	大年 倉像	"	武富 大児
"	井上 通	"	瀬辺 国昭
"	立川 清兵衛	"	倉橋 浩一郎
"	吉田 太郎	"	小原 英一
"	野畑 昭夫	"	尾形 善弘
"	田中 克介	幹事	岡田 俊介
"	的場 徹	6号特集担当	山田 郁夫
"	野村 兼八郎		

三菱電機技報 58 巻 6 号

(無断転載を禁ず) 昭和59年6月22日印刷
昭和59年6月25日発行

編集兼発行人 岡田 俊介
印刷所 東京都新宿区市谷加賀町1丁目12番地
大日本印刷株式会社
発行所 東京都千代田区大手町2丁目6番2号 (〒100)
菱電エンジニアリング株式会社内
「三菱電機技報社」Tel. (03) 243 局 1767
発売元 東京都千代田区神田錦町3丁目1番地 (〒101)
株式会社 オーム社
Tel. (03) 233 局 0641(代), 振替口座東京 6-20018
定価 1部500円送料別(年間予約は送料共6,700円)



このたび三菱電機では、パワーエレクトロニクスの新世代をひらく電力用ゲートターンオフ (GTO) サイリスタのシリーズ化を他社に先駆けて完成しました。

このGTOサイリスタは、従来の高速サイリスタにかわって産業用インバータや鉄道車両用インバータ、チョップ装置などに使われようとしています。このニーズにいち早く対応するため、 $<100\text{A}\cdot 1200\text{V}>$ から $<1800\text{A}\cdot 2500\text{V}>$ まで8品種にわたるシリーズ化を達成したものです。GTOサイリスタは、通常のサイリスタと同じターンオン能力と、ゲートに逆バイアスを与えるとターンオフするというトランジスタに似た自己ターン能力—この2つの能力をもっています。従って、インバータやチョップ装置にこのGTOサイリスタを採用することにより、これまでの転流回路が不要となり、装置の小形・軽量

化、高性能化、低騒音化を同時に実現することができます。このたびのシリーズ化により、GTOサイリスタの応用分野は一段と大きく広がるものと期待されています。

特長

- ターンオン、ターンオフ時間が速く、インバータ、チョップ装置の高性能化を図りました。
- 100A、200Aタイプ (GM100DY、GM200DY) は、GTOサイリスタおよび高速形のフライホイールダイオードを2組、1つのパッケージに内蔵しており、しかもこれらは冷却用ベースとは絶縁されているモジュールタイプとなっています。従って、主回路の配線および素子の冷却はきわめて簡単に行えます。
- 450A以上は、冷却効率のよい両面冷却タイプ・平形パッケージとなっており、きわめて高い信頼性を実現しました。

主要定格特性

項目	単位	形名	GM100DY	GM200DY	FG450BL	FG600AL	FG600AH	FG1000AL	FG1000AH	FG1800AH
ピーク繰返しオフ電圧 (VDRM)	V		800~1200	800~1200	800~1200	800~1200	2000~2500	800~1200	2000~2500	2000~2500
ピーク繰返し逆電圧 (VRRM)	V		15	15	15	15	15	15	15	15
可制御オン電流 (ITGQ)	A		100	200	450	600	600	1000	1000	1800
	Cs	μF	0.1	0.47	2	2	2	3	2	4
	Ls	μH	0.1	0.1	0.4	0.3	0.3	0.3	0.3	0.2
実効オン電流 (IT(RMS))	A		31	70	200	270	270	450	450	860
サージオン電流 (ITSM)	A		400	500	2500	5000	5000	7000	7000	11000
臨界オン電流上昇率 (di/dt)	A/ μs		200	200	200	200	200	200	200	300
オン電圧 (VTM)	V		4.5	4.3	2.2	2.2	2.5	2.2	2.5	3.0
ゲートトリガ電流 (IGT)	A		0.4	0.9	1.0	2.0	2.0	3.0	3.0	4.0
ゲートトリガ電圧 (VGT)	V		1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0
ターンオン時間 (tgt)	μs		4.0	4.0	10	10	10	10	10	10
ターンオフ時間 (tgo)	μs		8.0	10	15	15	20	15	20	30
	$-di_G/dt$	A/ μs	15	15	20	20	20	20	20	30
ターンオフゲイン (GGQ)	—		5	5	5	5	5	5	5	5
接合温度 (Tj)	$^{\circ}\text{C}$		-40~125	-40~125	-40~125	-40~125	-40~125	-40~125	-40~125	-40~125
熱抵抗 (Rth(j-θ))	$^{\circ}\text{C}/\text{W}$		0.7	0.35	0.09	0.055	0.055	0.045	0.04	0.024
形状	—		絶縁形2素子入	絶縁形2素子入	平形	平形	平形	平形	平形	平形