「会話型によるソフト開発の生産性向上例」

㈱立石電機
児玉　清

Ⅰ　はじめに

近年マイクロプロセサ（以下ＭＰＵと略す）を応用した装置商品の増加とともに，それを働かせるためのソウトウェアの開発量が増大し，ソフトウェア開発（以下ソフト開発と略す）における生産性向上が大きくクローズアップされてきた．

生産性向上のポイントを要約すれば次の３点にしぼられるだろう．

1）プログラム開発技法の適用

構造化プログラミングとかトップダウン開発のようなソフトウェアの設計方法，設計思想についての技法である．

2）ソフト開発システムの開発

ソフトウェアを開発するための道具を体系化し，まとめたものである．

3）使用言語の選定

ソフトウェアを構築する時に使用する言語で，コンパイラ言語とアセンブラ言語がある．

ここではソフト開発システムに関して，会話型によるソフト開発の生産性向上について述べてみたい．

当社では，昭和５２年８月に会話型ソフト開発システムの開発を完了し，現在までバージョンアップを繰返し，ＭＰＵソフト開発の生産性向上に大きく貢献している．

Ⅱ　ソフトウエア開発増加の背景

ＭＰＵ内蔵の装置商品をとりまく環境について考えてみると以下のことが言える．

1）最近のハードウェアの技術革新はめざましく，４ビットから１６ビット系までのＭＰＵが市場に出揃い，その適用範囲が拡大したこと．そして，それに伴ってＭＰＵのコストが下がってきたこと．

2）装置商品に対する市場の要求が多様化と高度化の方向にあること．さらに，その要求を満足させるため，装置商品のソフトウェア化が進み，それの占める割合が増大したこと．

3）したがって装置商品をうまく機能させるソフトウェアにもまたプログラム本数の増加とプログラムサイズの拡大という形で要求の多様化と高度化の影響が現われてきたこと．およびソフト開発が人的思考作業に大きく依存すること．

このようにソフトウェアの開発量増加とそれに伴う要員の増加傾向により，ソフト開発の生産性向上が，非常に大きな問題点となってきた．

Ⅲ　ソフト開発における生産性向上への動き

3.1　必要性の認識とユーザからの要請

前記の背景からソフト開発期間の短縮とその管理手法の確立，ソフトウェアの信頼性，および保守容易性の向上，の必要性が認識されてきた．また，将来的には図１に見られるように，人件費のソフト開発費全体に占める割合が増加すると予測されている．このような予測から効率的なソフト開発の重要性が認識され，ユーザ部門からシステム開発の要請が強く投げかけられた．

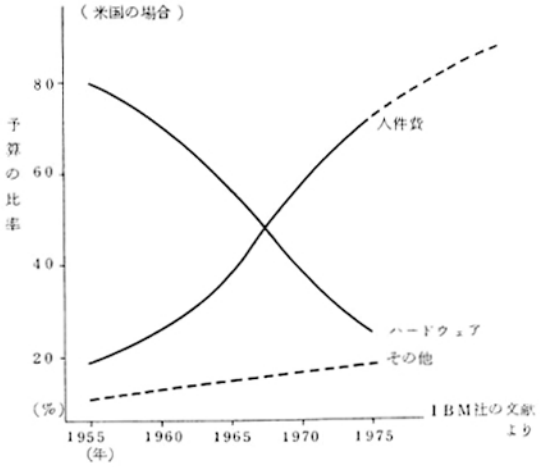

図　1

3.2　生産性向上へのアプローチ

このようなユーザの要請から，当社としてのソフト開発システムが開発された．このシステムは，当社のある装置商品を対象としたバッチ処理方式によるものであった．このソフト開発システムは次のような効果をもたらした．

1）マクロ展開機能と高級言語機能によって，コーディングステップ数を減少させ，２０～３０％の向上

効果が得られた．

2）さらに，連係編集機能によって，モジュール化を促進し，プログラム開発本数とテスト回数を減少させ２５～３０％の向上効果が得られた．

このような効果が得られたものの，納期短縮については，当初予定したほど大きな効果は見られなかった．その理由としては，バッチ処理方式に起因するところのコンピュータ処理による"待ち時間"を必要とし，それによる，"思考の断続"が発生したことが主たる原因と考えられた．

Ⅳ　会話型ソフト開発システムの開発

4.1　開発動機

会話型ソフト開発システム（以下本システムと略す）の開発を決心した理由は下記の３点である．

第一は前記アプローチ結果，納期短縮上からはバッチ方式の限界を感じたこと．そしてユーザからさらに効率的なシステムの開発要請があったこと．

第二はコンピュータ利用技術としてのＩＣＳ（Interactive Computing System）の技法を耳にした．その技法はソフト開発効率化に利用し得る可能性を感じさせた．

ＩＣＳの技法導入として，当社のハードウェア，その他の諸条件からＶＭ／ＣＭＳの採用が望ましいと判断した．

第三にはＶＭ／ＣＭＳにより，当社独自のソフト開発システムを構築できる確信が持ち得たし，また，それによってさらに生産性向上の期待が予測できたこと．参考までに図２にバッチ処理と会話型処理の比較を示す．

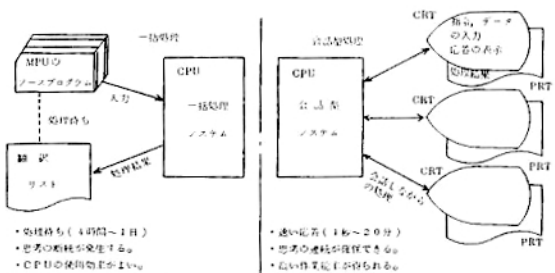

図２　一括処理と会話型処理の比較

4.2　開発目標

本システムの開発に当って，まずユーザに対しきめ細いアンケート調査を実施した．その結果ユーザからの要請は大よそ以下のとおりであった．

1）ソフト開発をさらに効率よく行いたい．

2）ＭＰＵソフトを簡単にデバッグできる機能がほしい．

3）簡単に使いこなせるコンピュータシステムがほしい．

4）自分自身で仕事のスケジュールをしたい．

この要請を受けて，"機能の向上をはかること，および，手軽に利用できるシステムであること"の２点を開発目標と決め，本システムの開発に着手した．バッチ処理方式に於けるマクロ化，高級言語化，モジュール化に以下の機能を追加することにした．

1）すでに登録されているマクロとかプログラム・モジュールを検索して手軽にその内容を見ることができること．

2）処理スピードを上げること．

3）デバッグ機能を提供すること．

4）処理結果を充実し多角的な利用に備えること．

5）遠隔地からも本システムを利用できること．

4.3　開発日程

図３で示すように本システムの開発は昭和５１年８月より１年間で開発を完了させるという目標を設定した．本システムを完成させるに必要な作業テーマとしては，

1）リロケータブル・マクロアセンブラ（以下アセンブラと略す）の会話型への移行

2）シミュレータの開発

3）ＶＭ／ＣＭＳの導入

など，これら３つのテーマを並行して進めねばならなかった．その上，ＶＭ／ＣＭＳについては初めてのことであり，かつ，邦文マニュアルが少なかった事など非常にきびしい日程となった．が，しかしプロジェクトメンバーの努力と上司の適切な指導により上記目標を満足する当社独自のシステムを予定どおり開発着手からわずか１年たらずで完成させることができた．

ここでＶＭ／ＣＭＳについて簡単に説明しておく．ＶＭとは仮想計算機システム（Virtual Machine Facility）の略で１台の計算機を時分割と装置分割

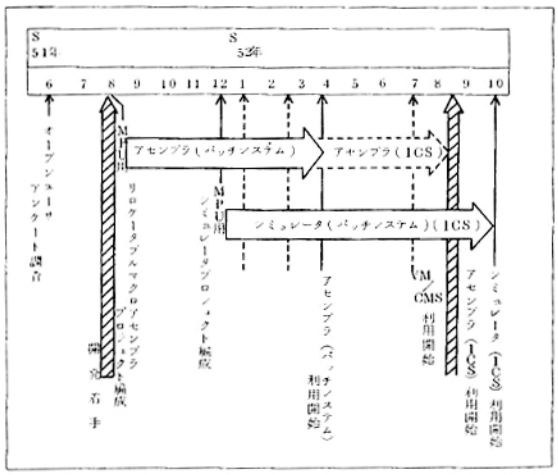

図３　開　発　日　程

することによって，複数台の仮想計算機を実現したシステムであり，ＶＭのもとでＣＭＳをはじめ様々なオペレーティングシステムを稼動させることができる．またＣＭＳ（Conversational Monitor System）とは会話型オペレーティングシステムの意味で，会話型に適した制御プログラムである．

4.4 開発体制

前述のように開発日程が短かかった事，会話型システムの開発が初めての試みで，数々の困難が予測できた事から，ＩＰＴ（Improved Programming Technique）の手法を適用し，プロジェクト制にて開発を進めることにした．

本プロジェクトにおいて，ユーザを含めたチーフプログラマチーム（図４），ウォークスルー，トップダウン開発，ＨＩＰＯ手法，ＳＰ（ストラクチャード・プログラミング）コーディングなど，ＩＰＴを取入れることにより，かなりな成功をおさめた．

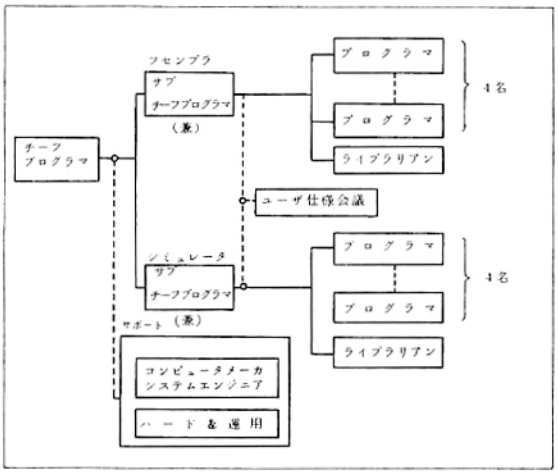

図４ チーム編成

開発を着手した当初，ＨＩＰＯによるドキュメント記述にかなりな時間を要した事，チームとしてユーザが参画したことにより，足並みに乱れが生じたことを記憶している．しかし，本システムが完成した現在，逆に以下のような効果として現われた．ＨＩＰＯによるドキュメントはその修正作業を減少させた．そしてプロジェクトにユーザが参画している事で，本システムのスムーズな移行が可能となり，利用拡大に大きく貢献する結果となった．

Ⅴ 会話型ソフト開発システムの概要

図５に本システム構成を示す．

以下のような６つの機能に大別できる．

1) ＳＴＯＲＥ（登録）

マクロやプログラム・モジュールをライブラリ（図中Maclib Pgmlib ）に登録するシステムで，登録年月日，作成者，課等の情報も合わせて登録できるようにした．以後，それをキイとして用い，そのキイが一致しないと，ライブラリに対して更新ができないようにし，ライブラリの保全を計った．

2) ＶＩＥＷ（検索）

キャラクタ・ディスプレイ（ＣＲＴ）から指令を入れることで，登録されたマクロやプログラム・モジュールを検索し，利用できるようにした．

これにより，ソフトの重複開発防止と標準化の推進の基礎ができ上がった．

3) ＡＳＭ（翻訳）

本システムの中心的な機能である．ＭＰＵのソースプログラム（図中Ｓｏｕｒｃｅ）を翻訳し，オブジェクトモジュール（図中Ｏｂｊｅｃｔ）を出力する．最もよく利用されることから機能の向上と処理速度の向上という点に特に努力をはらった．

4) ＬＩＮＫ（連係編集）

上記ＡＳＭで出力されたオブジェクトモジュールとすでに登録されたモジュールを連係編集して，完成されたロードモジュールを出力する．これにより，モジュールの再配置が可能となり，モジュール化を促進させた．ユーザは新規モジュールについて開発し，小さなプログラム単位の開発だけを行えばよく，実績のあるモジュールはライブラリから取出し連係編集して，全体を構成すればよいようにした．

5) ＳＩＭ（シミュレータ）

上記４つの機能で一応完成されたロードモジュールを得ることができるが，その中には論理的な誤りが含まれている可能性があり，文法的なチェックが完了しているにすぎない．このステップは一応完成されたロードモジュールを疑似的に実行して，論理的な誤りを見つけることを目的としている．

もし論理的な誤りを検出できた場合，ユーザの意思で，直接ロードモジュールを修正できるようにした．修正後，シミュレートを続行し，論理的にも完成されたモジュールを作り上げ次の機能へその出力結果を渡す．

6) ＯＵＴＰＵＴ（出力）

この機能は，ＭＰＵに接続されているメモリへ，完成されたロードモジュールを読込ませるため必要な種々の出力形式での出力を担当するものである．通常は紙テープへ出力されるが，回線を通じての遠隔地の種々の端末または装置商品そのものへの出力（センタローディング）を可能にしている．

このような６つの機能から成るシステムを開発して，前述した開発目標を達成することができたわけであるが，本システムの完成以後現在までユーザの意図を反映して数回のバージョンアップを行っている．

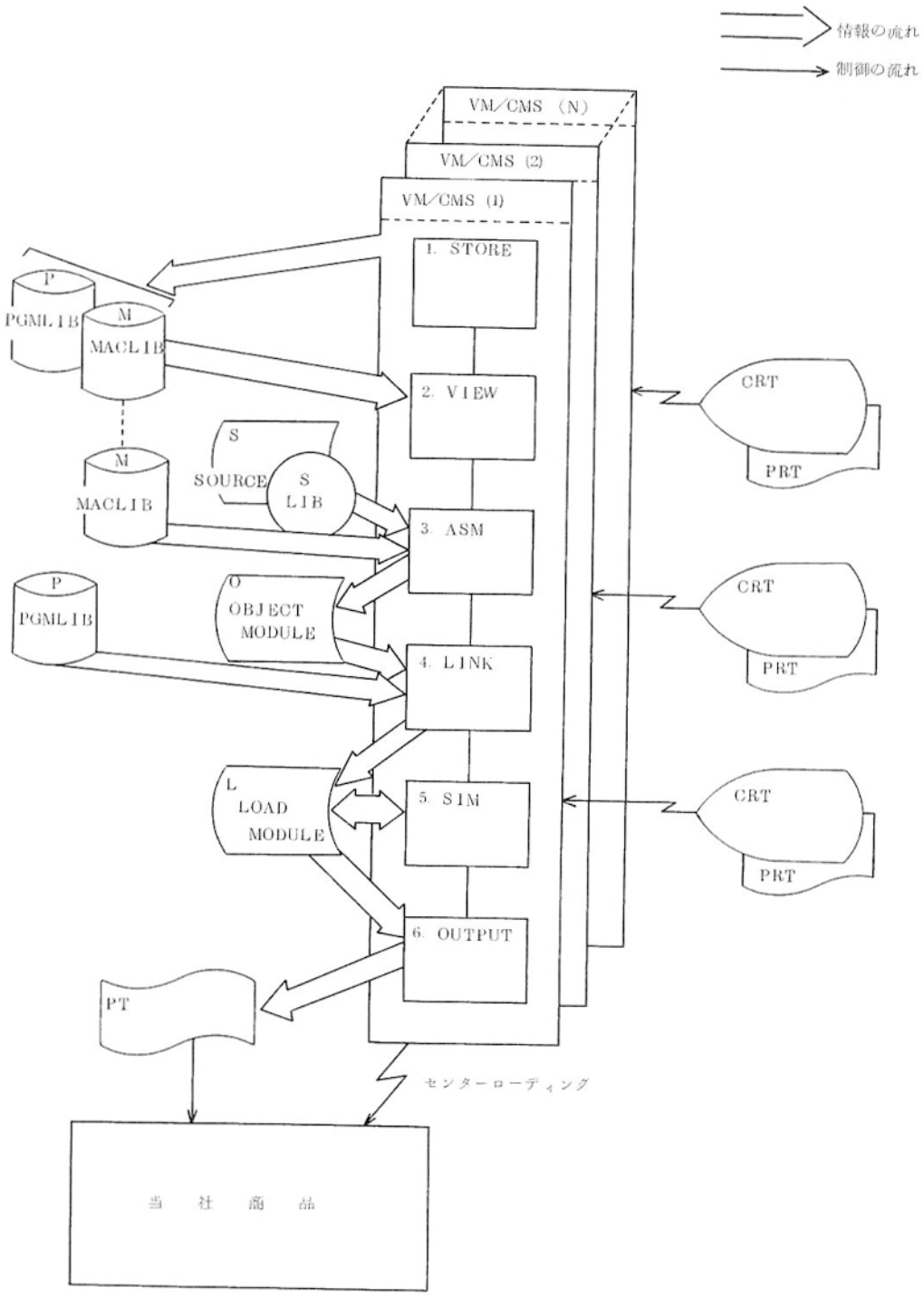
情報の流れ
制御の流れ
VM/CMS (N)
VM/CMS (2)
VM/CMS (1)
1. STORE
2. VIEW
3. ASM
4. LINK
5. SIM
6. OUTPUT
P
PGMLIB
M
MACLIB
M
MACLIB
S
SOURCE
S
LIB
P
PGMLIB
O
OBJECT
MODULE
L
LOAD
MODULE
PT
CRT
PRT
CRT
PRT
CRT
PRT
センターローディング
当 社 商 品

図5 システムの構成

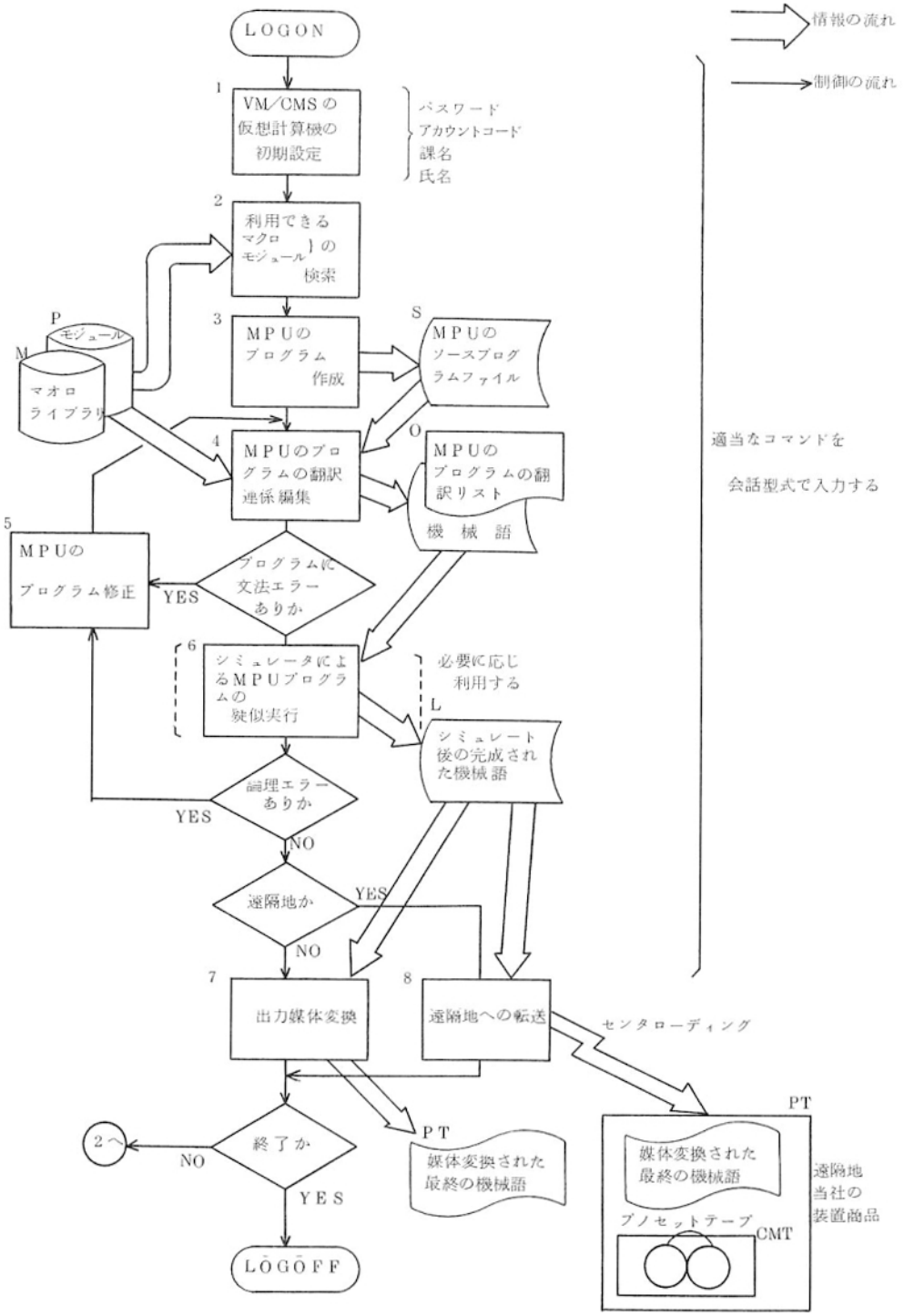

図6. 会話型によるソフト開発の実際

図6に本システムによるソフト開発の実際を示す．

ユーザはLOGON（作業の始まり）して，①パスワード（機密保護のためのコード），アカウントコード（計算機利用のための会計コード），課名，氏名等を入力すれば，LOGOFF（作業の終了）まで，②から⑧の作業に応じて，適当なコマンド（本システムに対する指令）を入力してゆけばよい．

Ⅵ　会話型によるソフト開発の効果

本システムの効果を列挙すると

1）機械処理の待ち時間が少なくなったことによる思考の連続によりソフトの生産性が（バッチ処理方式の効果を含めて）100%～150%向上した．

2）登録システムの開発により，部門または個人単位のノウハウをライブラリに蓄積し，全体として活用できるようになった．

3）シミュレータの開発により，開発段階で論理の確認が可能となり，プログラムの品質が向上した．そして，装置（ハードウェア）の完成を待つことなくソフト開発完了ができるという，言わばハードウェアからの独立が実現した．

4）本システムの基本ロジック部分を従来からのコンパイラ言語でなくアセンブラ言語で開発したため処理効率が向上し，より速い応答を実現できた．

Ⅶ　会話型によるソフト開発の有効性

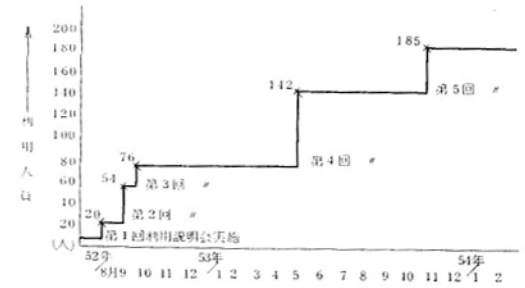

図7.　利用人員の増加

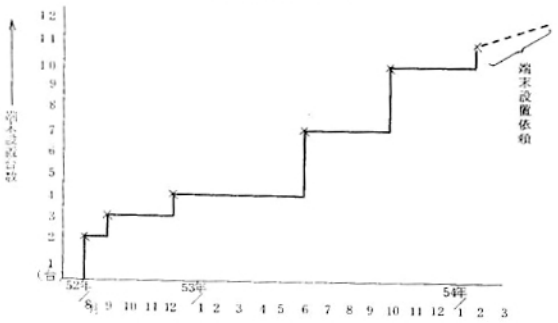

図8.　ＩＣＳ端末増設

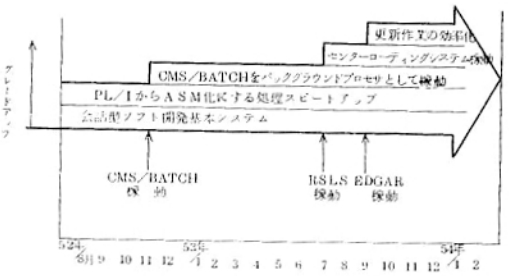

図9.　会話型ソフト開発システムグレードアップ

図7に利用人員の増加を，図8にＩＣＳ端末の増設を，そして図9に本システムのグレードアップの現状を示した．図9で見るように，より生産性を上げるための改善を実施してきている．

会話型で処理する必要のない処理をバックグラウンドで行わせるために1つのＶＭにて，CMS／BATCHを稼働させ，全体的な効率アップを計った．そして，RSCS（Remote Spoolting Communication Subsystem）を稼働させ，遠隔地への出力結果転送を開始した．さらに，EDGAR（ファイル編集プログラム）を導入して，ファイルの編集作業の能率を向上させた．

図7，図8にみられるように，端末機の各部門への設置に伴い，利用人員が増加し，約200名のユーザ全員が本システムを利用する日も近い．

Ⅷ　おわりに

前述したごとく，本システムの開発目的はまず達せられたと考えている．図8で示すように各ユーザ部門からのＩＣＳ端末設置の要望がますます大きくなり，端末数も増加傾向の一途をたどっている．これはユーザにその効果が認識され，さらにそれが拡がっていると言える．

さらに，本システムの開発体験を基にして，ソフト生産性をさらに向上させるための汎用クロスアセンブラシステムの開発完了も間近に迫っている．

今後一層使いがってのよいシステムを開発し，さらに生産性向上という限りないテーマにアプローチしてゆきたい．