

VM機能を持つTSSシステムの実現法

佐藤匡正 長野宏宣 (電電公社 横須賀電気通信研究所)

電電公社ではINS構想の具体化に伴い、高度かつ多様なソフトウェアの需要が顕在化してきている。パソコンをはじめとするスモールコンピュータ(SC)、電子交換機、DIPS等の多様な機種と、OSから応用プログラム(AP)に至る広汎な分野をカバーするソフトウェアを能率良く開発する必要に迫られている。これに対処する施策の一環として、研究所では多種多様なソフトウェアの開発を能率良く進めるための総合的な設備を構築中である。ここでは、本設備計画の概要を示すとともに、その実現上の主要な技術である仮想計算機(VM)機能を持つTSSシステム(TSS-VM)の実現方法について述べる。

1. 序

ソフトウェアの開発を能率良く進めるためには、作業環境を整備して設備として具体化することが重要である。

設備(開発設備)には、ソフトウェア工学ツールをはじめとするソフトウェア設備とハードウェア設備とがある。

環境(開発環境)では、作業自体を能率化するための支援と、作業の内滑な進捗を維持するための管理とを実現する。支援環境は更に、プログラム作成環境とソフトウェアシステム走行確認環境とに分けられる。環境の分類とその内容を表1に示す。開発設備には表1の環境が具備されていなければならない。

次に、設備としての具体的な条件を明らかにするために、開発の形態と環境との関係に着目する。今、開発の形態を便宜上、SC用のM型、メインフレーム用のAP, LP/UP, PKG, EXEC及びCMP(通信制御用プログラム)の各型に分類する。

表1 ソフトウェア開発環境の分類

開発のための環境	概要	内容
支援	プログラム作成	主として設計・製造に 関する作業を支援する。 エディタ, コンパイラ, 設計言語等のSE ツールキット
	システム走行確認	ソフトウェアシステムを走行 させるためのOSやパッケ ージ等が設定される。 多様な版のOS/パ ッケージ, 多機種の設 置
管理	開発を円滑に進めるため の管理作業を支援する。	管理情報用DB, データの自動収集 機構

表2 ソフトウェアの開発形態と開発環境

開発のための環境		開発形態						
分類	機能	M	AP	LP/UP	PKG	EXEC	CMP	
支援	作成	○	○	○	○	○	○	
	支援ツール	○	○	○	○	○	○	
	走行確認			○	○	○	○	
管理	異機種	○					○	
	版管理	○	○	○	○	○	○	
	アクティビティ管理					○	○	

これらの各型で必要とする開発環境を表2に整理する。表2から作成環境はほぼ同様な機能でよいが、走行確認環境は主として確認対象のソフトウェア体系における位置付けにより、大幅に異なることがわかる。

広汎な適用性を持つ設備を構築するには、このように多様な走行確認環境を具備することが重要である。

2. 総合ソフトウェア生産設備の概要

ソフトウェアの設計から保守に至るライフサイクルを総合的に支援するために、序に述べた三つの環境を具備した設備(WAVE, 図1)を構築中である[1]。

本設備は、プログラム作成を会話的に行うためのTSSセンタを中心に、機能分散・負荷分散をねらったBEP構成である。BEPは、開発経験の自動蓄積をはかるためのデータベースセンタ(DBセンタ)及びソフトウェアシステ

ムの動作確認を行うための走行確認センタ（VMセンタ）から成る。更に、各センタにはPATTYディスク（30GB）、MSS等の大容量ファイルを設置してセンタでの操作を極小化している。

開発ステーション（WS）には一般のパソコンをTSS端末として使用する方を確立した〔2〕。この他、DIPS Vシリーズや民需機等のミニコンをスーパーWSとして設備する。WSのインテリジェンスレベルに応じてテキストエディットから単体試験までの機能をセンタから分散させることができる。

3. TSS機能とVM機能との融合

3.1 背景

(1) 設備の不足

プログラムに誤りの無いことを保証することは一般に難しい。現実的には走行させてみて予想どうりの結果が得られれば、是としている。この際の走行環境は実際のものに近い程、好ましいとされている。特に、複雑な入力パターンや内部状態を持つプログラムでは、予期せぬ状況が生じ易く、動作不良の要因となることが少なくない。擬似的な走行環境に於ては、論理上の妥当性は確認できるが、現実には生じる例外的な処理、事前の処理からの引継ぎ等によって生じるタイミング上

の狂い等に関連する誤りが顕在化しないままに看過されやすい。従来はこの様な走行確認環境を実現するために計算機を占有的に使用してきた。ところが最近のネットワーク型システムの開発では、複数の計算機を同時に確保するために、膨大な計算機資源が必要となり、これに見合うだけの設備が準備できなくなっている。

(2) 分担開発

システム規模の巨大化に伴い、単一組織では担当し切れず、複数の組織による分担開発が採用されている。各組織は互に遠隔地にあると想定され、全体を独立なサブシステムに分割して開発する。開発の終わったサブシステムを改めて統合し全体を確認する。この様な開発形態が効率的に機能するためには、各組織の作業結果を他組織（支部）又は総括する組織（本部）へ即ちに提供でき、各組織の作業が円滑に連動して進行できる機構が必要である。

3.2 TSS-VM機能の設計方針

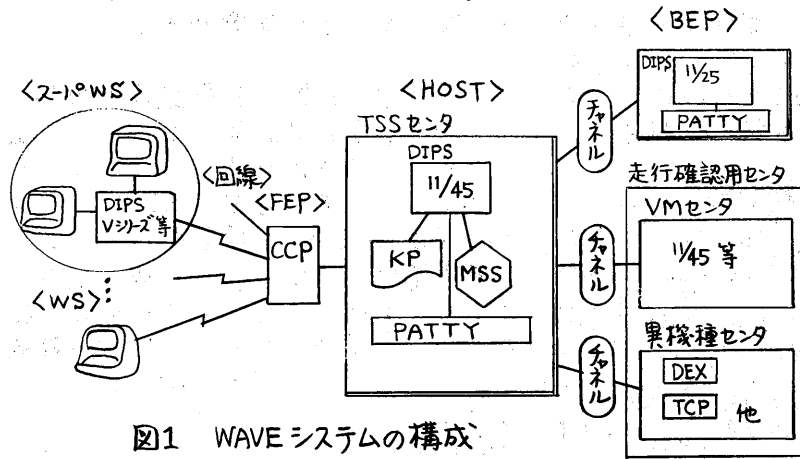
TSSシステムに具備されている計画立案支援、プログラム作成支援、走行結果の分析などの支援機能と組み合わせてVMによる走行確認機能が使用可能な環境を構築する。このための主要な設計方針を以下に示す。

(1) 操作性

TSSライブラリの1つとしてVMの操作機能を提供する。WSはTSS端末及びVM用仮想コンソールとして動的に切り替えられる。

(2) 性能・拡張性

機能分散、負荷分散構成がとれるものとする。



(3) 早期完成・建設費の削減
徹底して既存資産を流用し、性能等については高性能ハードウェア装置の配置によって解決をはかる。

3.3 構成と動作原理

(1) 構成概要

図1からTSS-VM部分を取り出して、システム構成とその上に搭載されたソフトウェア構成を図2に示す。TSSセンタにはWSに対応して会話処理タスク(CUT)とシステムに唯一つVM通信用タスク(RCT)が存在する。VMセンタには、VM用のモニタ(VMM)の配下にTSSセンタとの通信及びVMMとの通信を行うVM(SVM)がある。更に、CUTと対応する走行確認用のVMがある。

(2) 動作原理(図3)

今、WSからVMライブラリを通じてVM起動要求を上げたとする。この要求はCUT_iを介してRCTのキューにつながる。SVM上にはTSSセンタと同一のEXECが存在しており、チャンネルを經由してRCTからの要求を受け取る。この要求はSVMからハンドシェイク機構によってVMMに通知され、VM_iが起動され

る。起動完了の通知は、VM側から逆のルートを経由してWSに知らされる。この時点からWSはVM_iのコンソールとなる。このコンソールからは通常のセンタ運用コマンドによる操作が可能となる。この他に、VMMへの指示やTSSモードへの復帰等ができる。VM_i上での走行確認のためのテストデータ、処理結果、ダンプ等はファイルに格納され、VM処理前後または中断点でTSS機能を利用して作成・参照する。

3.4 主要な機能

(1) VM操作機能

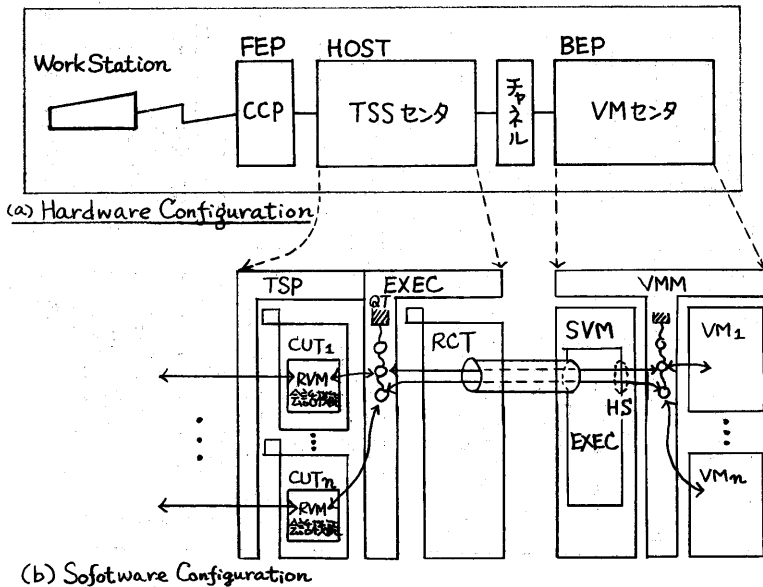
WSからのVMの開始・終了の他に、中断・再開を可能とする。又、デバッグ支援用にVMMとの通信機能を持つ。

(2) スプール機構

仮想CR/LP装置によって、予め必要となる入力情報の設定、及び出力結果の保存を可能とする。

(3) LCMP機構

VMで使用するファイル類はLCMP機構によりTSSセンタからも、VMセンタからも作成・参照を可能とする。本機構によって、テストデータの準備、



EXEC: 制御プログラム
TSP: TSSパッケージ
CUT_i: 会話ユーザタスク
RCT: BEP通信用タスク
VMM: VMモニタ
VM_i: 走行確認用VM
SVM: ホスト通信用VM
HS: ハンドシェイク

図2 TSS-VM機能の構成

VM 処理結果のファイルやダンプ等の参照がTSS センタでも可能となる。

3.5 TSS操作と組合せたVM操作法

WAVE の操作はメニュー選択方式を基本としている。図4に示す様に、試験走行でVMのメニューを選ぶと、次々に画面が切り替り、起動のために必要なパラメータの入力要求画面が表示される。VMが起動された後は、通常のコンソールと同様にVMの操作が可能となる。

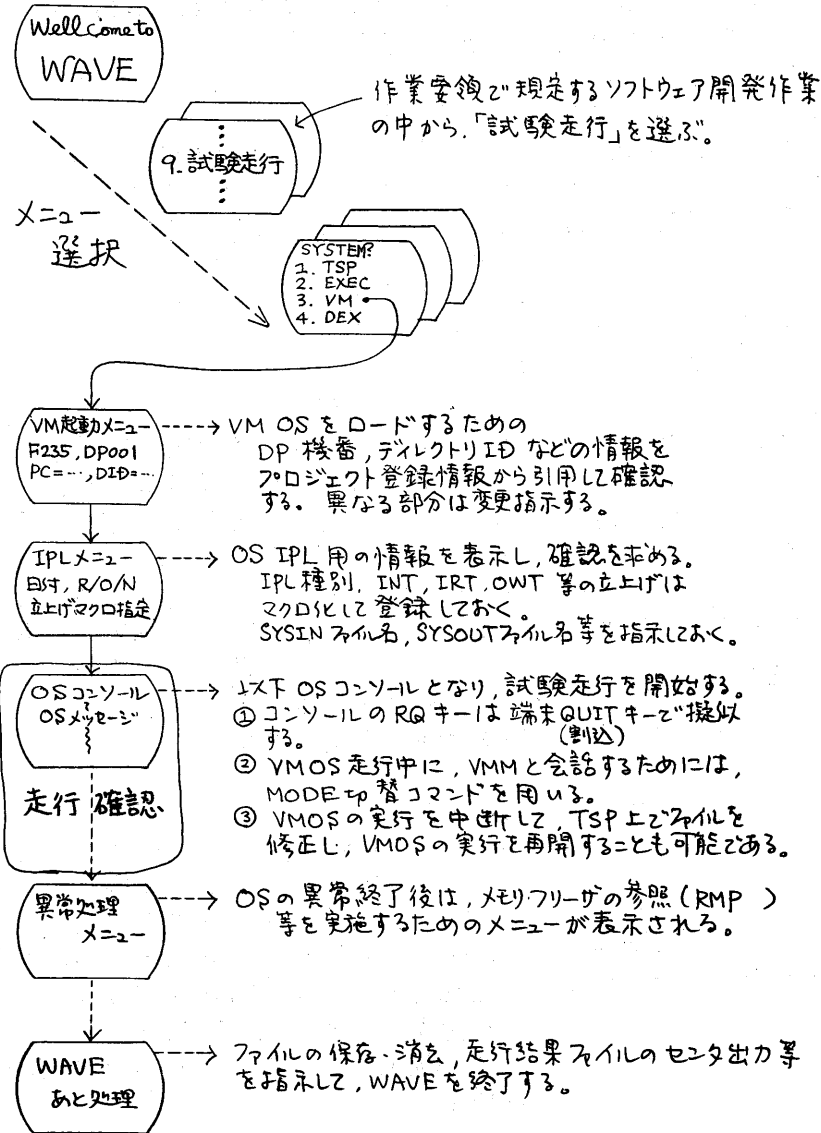


図4 TSS操作と組合せたVM操作法

なる。

4. 使用例

本システムの有効性を示すために、TSS機能とVM機能を効果的に組合せてシステムプログラムを開発する場合を例にとる。

開発組織は図5に示す様に、本部と複数の支部から成る。支部では単体、及びサブシステムを作成する。本部では各支部で開発したサ

ブシステムを統合してシステムを組み上げ、試験を行う。

以下の作業ステップで作業が進められる。

図6にこのモデルを示す。

以下①はTSS機能②はVM機能を用いることを示す。

S1: 走行確認計画の立案

サブシステムを単位とする確認順序・期日を示すアクティビティ・ネットワーク(ANW)を作成する。①

作成過程で支部の意見が反映される。①

S2: 走行形式プログラム(LM)の提出

S1の計画に従って支部では単体・サブシステムを作成する。①

ANWの期日に従って、本部は支部にLMの提出を指示する。① 支部ではLMを作成し、所定のDBに格納する。①

S3: ミシステムジェネレーション(SG)

本部では走行確認用ファイルを作成する①。SG結果の出カリスト等はセンタファイルに格納されており、各支部・本部で参照される①。

S4: 走行による動作確認

図4に示した操作によってVM_iが起動され、S3で示したファイルがロードされる①。予め作成しておいた①処理対象データファイルやJCL、及びWSによるコンソール操作によって必要とするプログラムを走行させる①。VM処理後にはWSは元のTSSモードに戻り、処理結果、ファイルダンプ等により動作を確認する①。

S5: 確認OKの場合

対応するアクティビティについて走行確認済みをもANW上にマークする①。

S6: トラブルの処置

SS1: 不都合な箇所(トラブル)が検出されれば、本部で一次解析を行い、疑惑のある箇所を確定させる①。結果を担当の支部へ通知する①。

SS2: 支部では保存されている処理結果、ダンプ等の検索①、VMを用い

た再現試験①、等によりデバッグを行う。

SS3: ファイルを修正して本部へ渡す①。本部ではS4を再実行し、修正の是非を確認する。

S7: 完了

所定の全アクティビティについて走行確認済みとなった段階で、ファイル統合完了となる。

なお、以上のステップでは説明を簡明にするために、信頼性保証活動の部分を省略してある。

4. まとめ

基本ソフトウェアの開発を支援するためには、充実した設備が必要である。従来のSEツールは応用プログラム分野が支援対象の中心であったため、走行確認環境を充実させるための設備という考え方が希薄であり、ソフトウェアによる支援に偏っていた。

我々は、「環境に対する設備」の考え方に立って、作成・走行確認・管理を一体化した総合的な設備を構築中である。本設備によってVM機能の有効性が更に促進されるものと考えている。

TSS-VMは58年8月に第1版を完成させる予定である。今後は、実際上の運用経験を積みながら、機能拡充をはかり更にその効果を向上させながら、全システムの構築を推進させる。

参考文献

[1]佐藤、長野他:「機能に着目したソフトウェアツール体系化の提案」、第25回情報全国大会、1982年10月

[2]佐藤、長野他:「端末操作の高水準化を狙ったシナリオ機構の提案」、第26回情報全国大会(投稿中)

[3]佐藤他:「基本ソフトの遠隔地開発環境」、同上

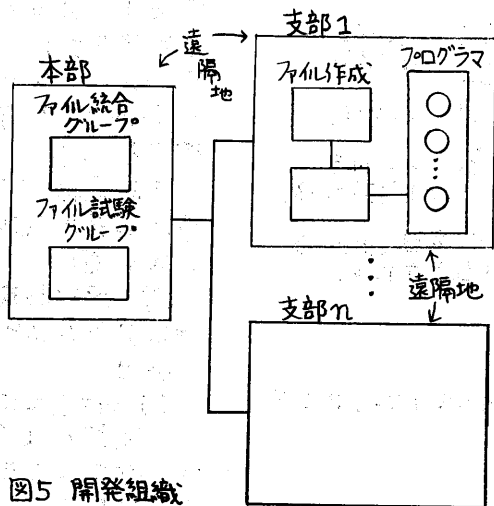


図5 開発組織

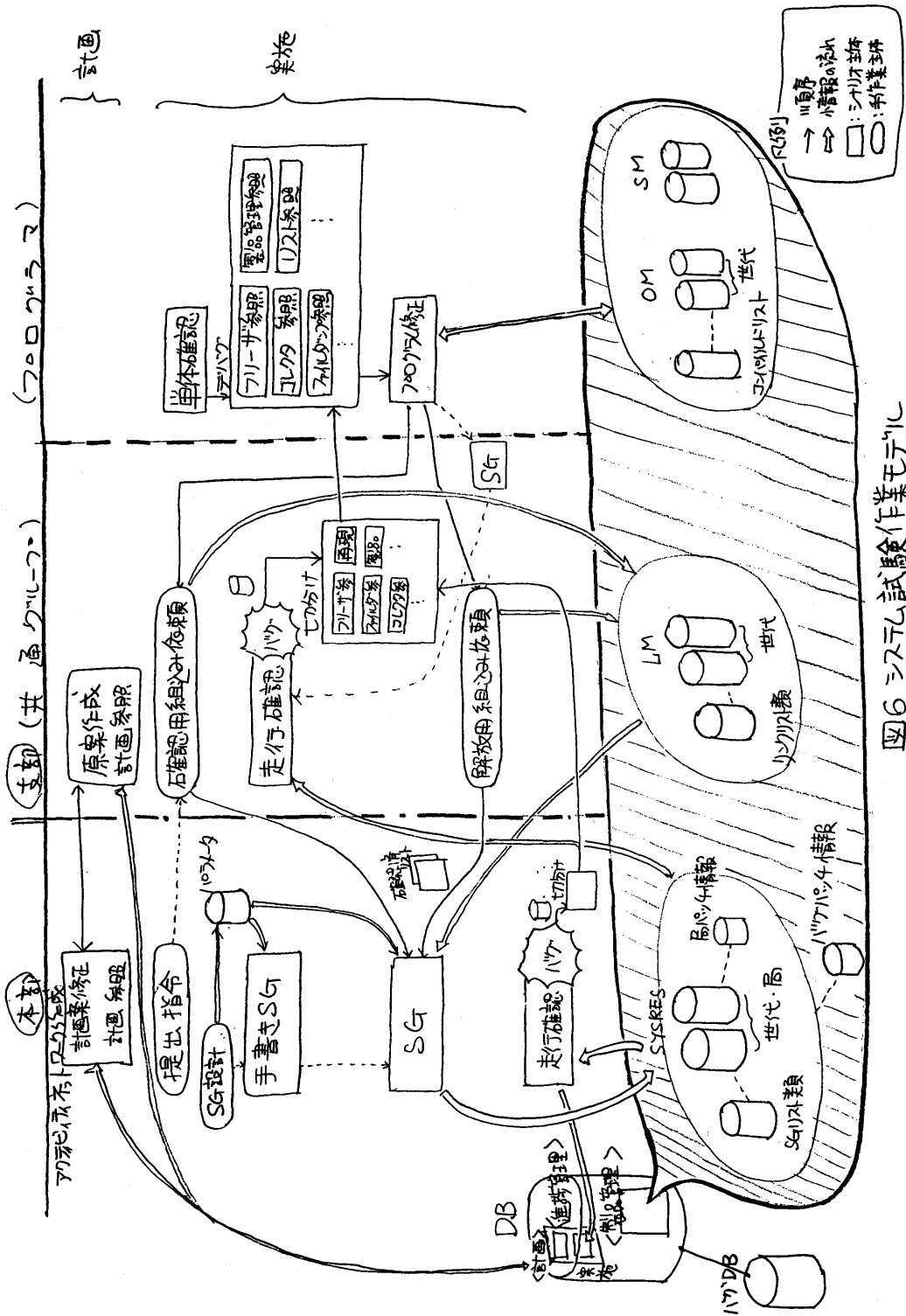


図6 システム試験作業モデル