

通信用ソフトウェア生産技術の高度化

Highly Efficient Support System for Telecommunication Software

日立製作所では、通信システム関連の機能の多様化、複雑化に伴い増えつつあるソフトウェアの開発、及び保守作業の効率向上とマイクロコンピュータ導入の活発化によるソフトウェア開発環境の変化に対処するために、ソフトウェアの開発から保守までのライフサイクルを通して一貫性のある高性能ソフトウェア開発支援システムの実用化を行なっている。

今回、実用化したものは、特にマイクロコンピュータ用の大規模ソフトウェアの開発に適用することを目的としてきたもので、ソフトウェアの各開発フェイズに対応した5個の支援機能から構成されている。

本システムは、各種通信システム用ソフトウェアの開発に使用され、品質及び生産性の面で大きな効果を挙げている。

黒崎 徹* *Tōru Kurosaki*
 本田明徳* *Akinori Honda*
 渡辺 坦** *Tan Watanabe*
 桧山邦夫** *Kunio Hiyama*
 小林正和** *Masakazu Kobayashi*
 浅田正彦*** *Masahiko Asada*

1 緒 言

近年、電子交換機システムをはじめとする各種通信システム用ソフトウェアの開発量、既開発ソフトウェアの保守作業が急増している。一方では、通信システムへのマイクロコンピュータの導入とそのソフトウェア規模の増大などがあり、通信システムを取り巻くソフトウェア開発環境が急変しつつある。これらに迅速に対応でき、柔軟性のあるソフトウェア生産技術を確立させ、ソフトウェアの品質及び生産性の向上を図ることが急務である。

ここでは、ソフトウェアのライフサイクルを通して一貫性のあるソフトウェア開発支援システムの概要と、これを構成している代表的な支援ツールの機能について述べる。

このシステムは、特にマイクロコンピュータを使った各種通信システムのソフトウェア開発に適用しているものである。

2 ソフトウェア生産技術の概要

2.1 高度化の背景と実用化方針

ソフトウェアの品質及び生産性の向上を図るためには、その設計、検査などの技法の改善、ソフトウェア開発要員の教育、標準化の推進及び開発支援システム整備の4本柱の同期を取りながら進める必要がある。

しかし、マイクロコンピュータのソフトウェアに関しては、前述の各施策がうまく同期していない面があった。

4ビット、8ビットから16ビットのマイクロコンピュータの出現により、そのソフトウェアは、部品からシステムレベルへと大規模化し、複雑になってきた。このようなマイクロコンピュータ用大規模ソフトウェア開発の生産技術、特に開発支援システムについては、高性能のシステムを早期に実現する必要があった。高性能開発支援システムの実用化に当たっては、長期的視野に立っての生産技術の在り方、すなわち、ソフトウェアのライフサイクルを通しての一貫性のあるトータルシステムの構築を目指し、次に示すような実用化方針を立て実行に移した。

- (1) 通信サービス及びプログラム仕様の言語化と図式表現の実用化
- (2) リアルタイムシステム向き高級言語の開発

(3) シミュレータによるプログラムモジュールテストの標準化、及びテスト項目作成の効率化

(4) ソフトウェア開発管理情報収集及び分析の迅速化

これらの方針を実現させるためのホストマシンとしては、実システムのようなレジデント形は、入出力機器、メモリ容量及びオペレーティングシステム機能不足などの問題点をもっているため、TSS(Time Sharing System)オペレーティングシステム機能をもつHITAC Mシリーズをクロスホストマシンとして採用した。

2.2 ソフトウェア開発支援システムの構成

システム構成は図1に示すとおりで、設計、プログラミング、テスト、運用及び保守の各開発フェイズを支援する4個の支援機能と、各開発フェイズを横断的にみたソフトウェア生産管理を支援する機能の合計5個の支援機能から構成されている。

また、プログラムライブラリ及び生産管理用各種データは、それぞれ専用のデータベース管理用ツールによって管理されており、既述5個の支援機能の個々のツールはこれらデータベース支援ツールを介して情報の授受を行なうようになっている。以下に、各開発フェイズに対応した4個の支援機能について述べる。

3 設計支援機能

ソフトウェアの設計では、対象とするシステムの外部仕様(交換サービス)と、これらの外部仕様をプログラムで実現するための内部仕様(モジュール仕様)を明確に表現し、設計からプログラミングへの正確な展開作業を行なう必要がある。このための支援ツールとして次のものがある。

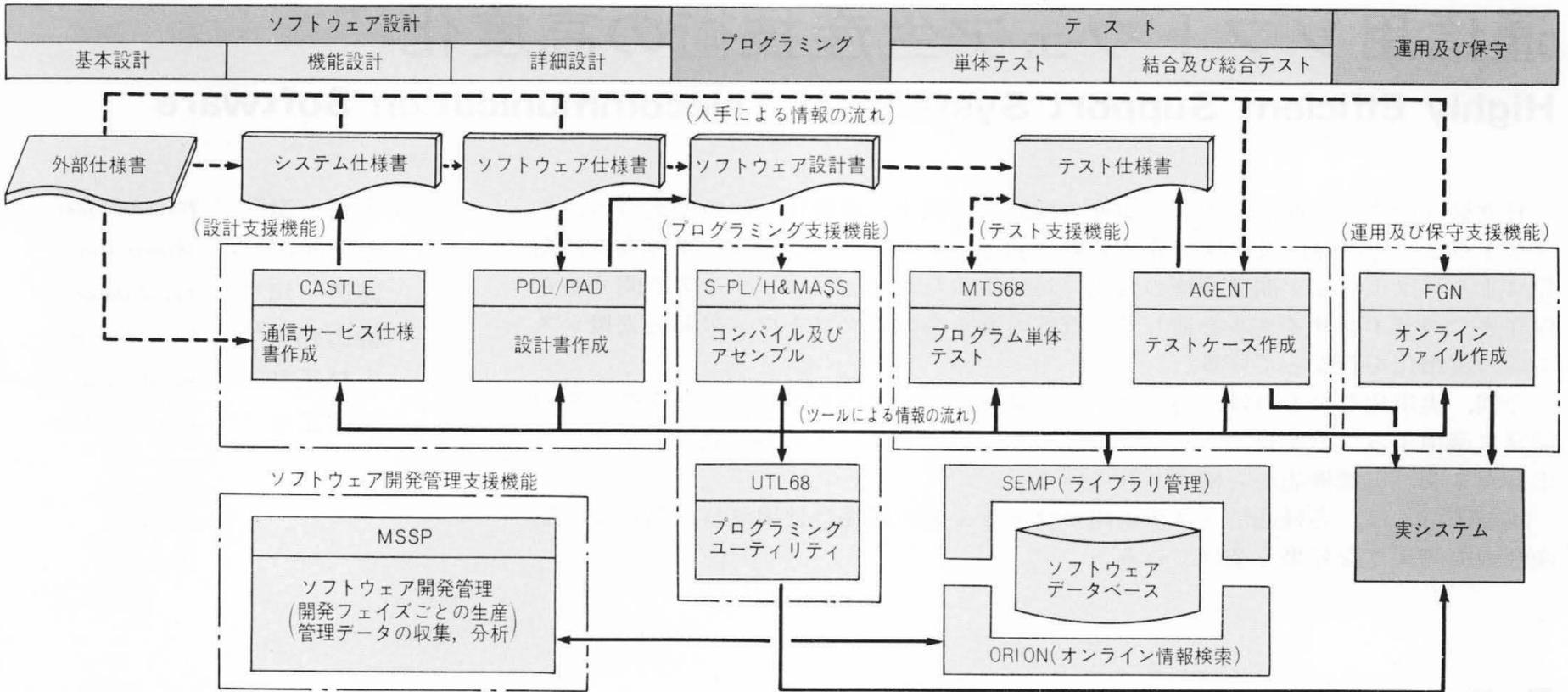
3.1 通信サービス仕様記述用ツール“CASTLE”

通信システムの処理プログラムは、呼びの処理を処理要求の監視(状態)と検出された処理要求の実行(状態の遷移)に分けている。例えば、交換機は、電話機やトランクの状態変化に応じて状態間を遷移するシーケンシャルマシンとして取り扱うことができる。

CASTLE(Call State Transition Language)は¹⁾、通信

* 日立製作所戸塚工場 ** 日立製作所システム開発研究所 *** 日立製作所武蔵工場

(ソフトウェア開発フェイズ)



注：略語説明
 CASTLE(Call State Transition Language)
 PDL/PAD(Program Description Language/Problem Analysis Diagram)
 UTL68(Utility for HMCS 68000)
 MTS68(Module Testing System for HMCS 68000)
 AGENT(Automated Generator of External Test-cases)
 FGN(File Generator)
 MSSP(Management System for Software Production)

S-PL/H & MASS(Super Programming Language & Macro Assembler for Hitachi Microcomputer System)
 SEMP(Source Code Editing and Maintenance Program)
 ORION(Online Retriever of Information)

日立製作所のプログラム製品でSEMPはLIMEという製品名で販売している。

図1 ソフトウェア開発支援システムの構成 マイクロコンピュータを利用した大規模ソフトウェア開発に使用されているツール群の全体を示したもので、5個の支援機能から構成されている。

サービス仕様を状態遷移で記述した呼び状態遷移とほぼ1対1対応で書ける言語で、CASTLEコンパイラは、この言語で記述したサービス仕様から状態内の差分を求め、処理要求を実行するためのタスクの自動生成を行なうことができる。また、図2に示すように呼び状態遷移図の自動生成を行なう機能も持っている。

3.2 プログラム設計用ツール“PDL/PAD”

ソフトウェア設計フェイズでの機能設計(モジュール設計)が終わった時点で具体的な機能の実現方法、すなわちアルゴリズムを明確にし、プログラミングフェイズに伝えるためには、アルゴリズムを論理的にだれもが容易に理解できるように表現を工夫する必要がある。

これを解決するために、2次元木構造でアルゴリズムを表現できるPAD(Problem Analysis Diagram)技法²⁾を導入した。また、このPAD技法の導入と合わせて、図3に示す支援ツールPDL/PAD³⁾を設計フェイズに活用してきた。

図3に示すように、アルゴリズム仕様は、一種の擬似コードPDL(Program Description Language)、又は図形形式により入力し、この仕様をもとに、PG(Program Generator)によりソースプログラムと書式を整えた仕様書の自動生成を行なう機能も持っている。また、テストフェイズでのアルゴリズムの修正、あるいは一括して大量のPADを作成する場合には、オートPADによりPADの作成を行なうこともできる。

4 プログラミング支援機能

ソフトウェア設計書から正確かつ迅速にプログラミングができ、各種テストのためのプログラム加工が容易にできるようなツールが必要である。このために、S-PL/H(Super Programming Language for Hitachi Microcomputer System)、マクロアセンブラなどの言語とこれら言語の周辺ツール群を実用化した。

4.1 高級言語“S-PL/H”

16ビットマイクロコンピュータの出現により、システムレベルの大規模ソフトウェアの実用化が可能になった。このソフトウェアの信頼性、生産性及び保守性の向上とシステム記述言語の標準化とを目的として開発したのが、HMCS 68000(Hitachi Microcomputer System)用高級言語S-PL/H⁴⁾とそのコンパイラである。

また、この言語の開発では、8ビットマイクロコンピュータ用高級言語PL/Hの上位移行性についても十分に考慮している。

特にこの言語は、制御向けプログラム及び構造化プログラミング機能を強化しており、S-PL/Hコンパイラ機能とし

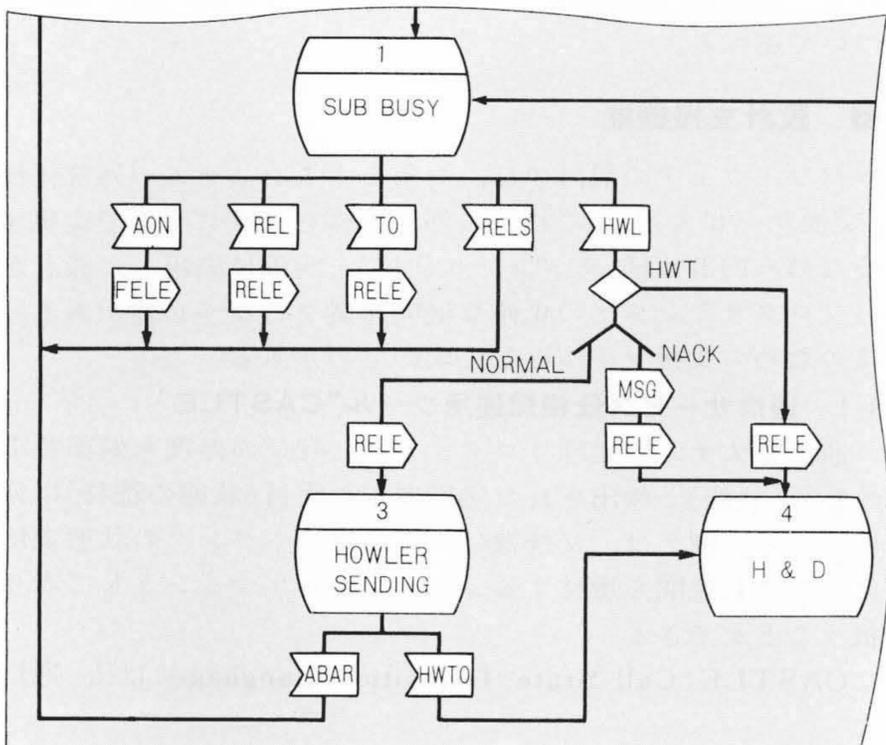


図2 状態遷移図出力例 電話交換機での電話端末の復旧監視機能をCASTLEコンパイラで出力したものである。

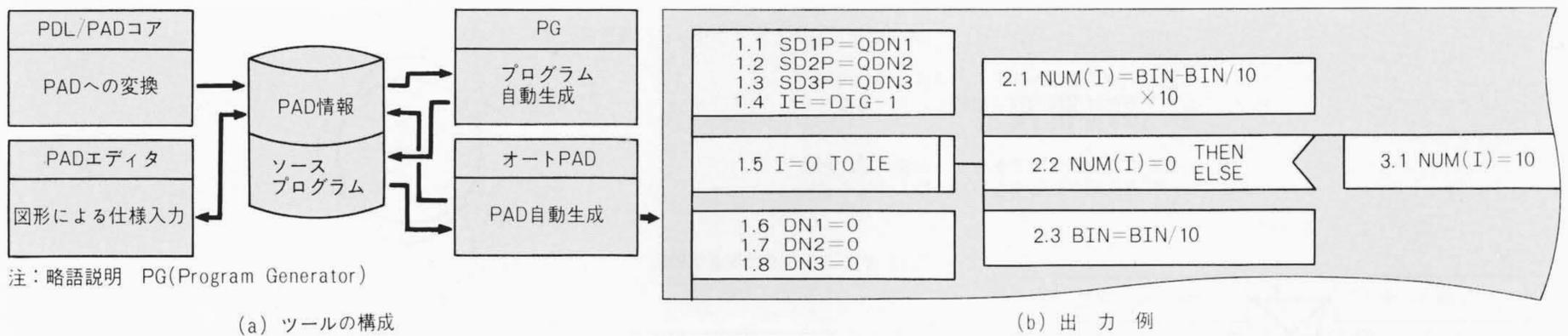


図3 PDL/PADの構成と出力例 プログラムモジュール仕様(アルゴリズム)を入力し、これの解析と設計書作成などのツールの構成及びオートPADによる出力例を示す。

ては、外部からの指定により目的プログラムのリエントラント化及びプログラムをROM(Read Only Memory)化した後で移動できる機能などももっている。

4.2 プログラミング支援ユーティリティ“UTL68”

実システムのファイル作成に用いられるツールとしては、ROM領域とRAM(Random Access Memory)領域へのメモリの分離割付けの容易なリンケージエディタ、クロスホストシステムと実システムのデータセット形式の変換を行なうものがある。

実システムでのテストを容易にするために、プログラムの部分的な索引が容易にできるように、便利ナリスト類を作成するツール群も用意している。

また、コンパイル、単体テスト、ファイル作成などのジョブ制御言語(JCL: Job Control Language)を生成したり、ソースプログラムの修正に伴って自動的にプログラム変更履歴情報及び工程進捗管理情報を収集する一連のTSSコマンドプロシジャ群を用意し、日常的なプログラミング作業の効率向上を図っている。

5 テスト支援機能

ソフトウェアのテストは、高品質のソフトウェアを生産するために極めて重要なフェイズである。そのため、テストの信頼性と効率の向上のために、以下の二つのテスト支援ツールを用意している。

5.1 プログラム単体テスト用ツール“MTS68”

コーディングされた各モジュールは、まず、単体でその機能をテストする必要がある。しかし、単体テストのためには大量のテストデータ及びテストプログラムを必要とし、テストごとにクロスホストシステムから実システムにプログラムを転送してテストデータを入力していたのではテスト作業の効率が悪い。このため、クロスホストシステム上でプログラムの単体テストを効率良く行なえるツールMTS68を開発した。

MTS68は、図4に示すようにテストデータ処理部と命令シミュレーション部とで構成される。テストデータ処理の入力部では、テスト条件記述をもとに被テストプログラムの入力テストデータファイルを作成する。このテスト条件記述は、入出力データの宣言、各テストケースの入力データ値、期待される出力データ値などで構成されている。命令シミュレーション部では、この入力テストデータと被テストプログラムモジュールを用いてシミュレーションを行ない、その結果をテストデータ処理部に報告する。テストデータ処理の出力部では、シミュレーション結果とテスト記述の中の正解値とを比較し、テスト結果判定リストを出力する。

MTS68を使用することにより、モジュールの単体テストを多数のプログラマが同時に効率良く行なうことができ、テスト作業の効率と品質を大幅に向上できる。また、テスト条件

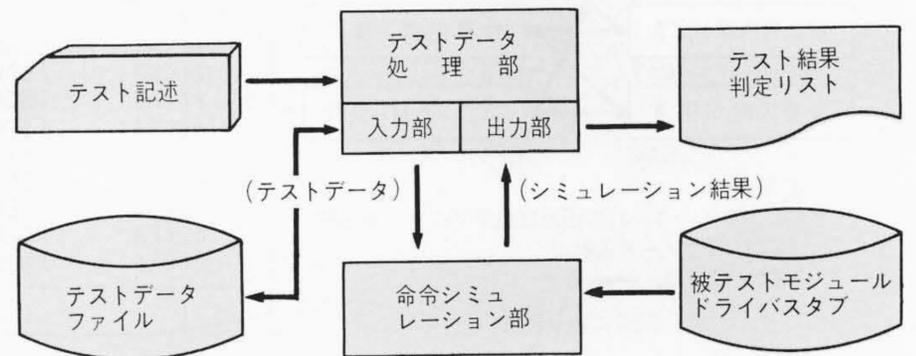


図4 MTS68の構成 プログラムモジュール単体テスト支援ツールの構成を示す。

記述を保存しておくことにより、システムの機能拡張時の再テストも効率良く行なうことができる。

5.2 システムテスト用ツール“AGENT”

外部仕様(システムユーザーからみた仕様)からのテスト項目の作成で、個人差がなく、質的に均一で、しかもむだのないテスト項目の自動生成を目的として開発されたツールがAGENT⁵⁾(Automated Generator of External Test-cases)である。

このツールの使用手順を示すと次のようになる。

- (1) 外部仕様をよく理解し、仕様書の機能を入力条件(原因)と出力条件(結果)とに分解する。
- (2) (1)の結果による論理記号を用いてCEG(Cause and Effect Graph: 原因結果グラフ)を作成する。
- (3) CEGをAGENT用の入力言語に変換することによって、テスト項目を自動生成する。

以上述べた使用手順を、具体的通信システムの一機能へ適用した結果を図5に示す。

6 運用、保守支援機能

電子交換システムのような通信システムは、顧客別及び用途別に要求機能が異なり、かつ多様化する交換サービス機能に対処するため、柔軟性の高いソフトウェア構造にしておく必要がある。図6に、交換ソフトウェアの構成と支援ツールFGN(File Generator)の関係を示す。

6.1 オンラインシステムゼネレータ“SGN”

各種要求機能に対して柔軟性の高いソフトウェア構造として新たに分散形状態遷移方式⁶⁾を開発し、機能との明確な対応がとれた機能別モジュール構成を実現した。SGN(System Generator)はこの結果を活用したもので、図7に示すように、共通ライブラリ作成部、モジュール選択部及び結合編集部から構成されている。機能的には、交換機能とモジュールの対応関係をデータベースに格納し、顧客要求機能を入力として目的とするOM(Object Module)を探し出し、顧客の要求機能に合ったオンラインシステムファイルを、効率良く、かつ正確に作成することができる。

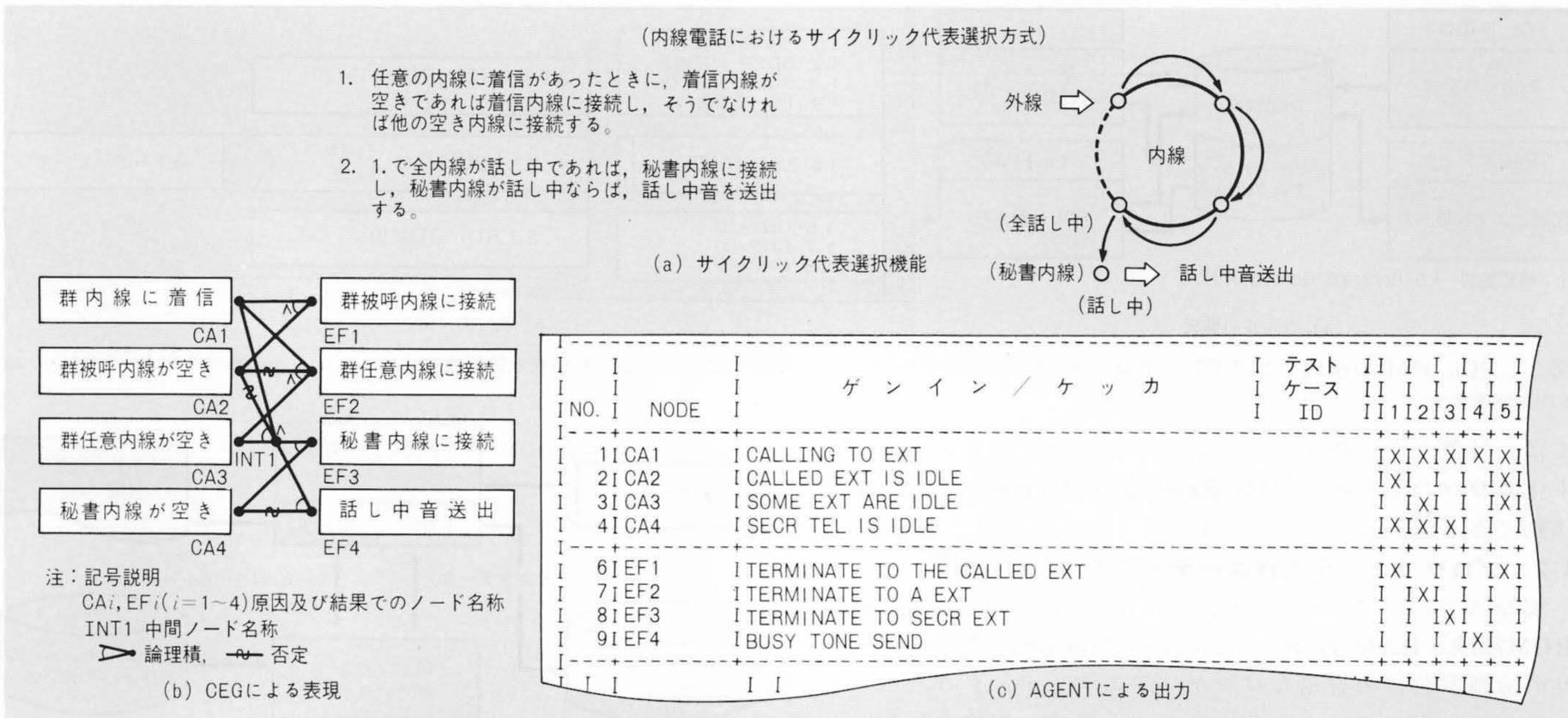


図5 AGENT運用手順例 私設電子交換システムの一機能について、運用手順に従っての記述及びAGENTの出力例を示す。

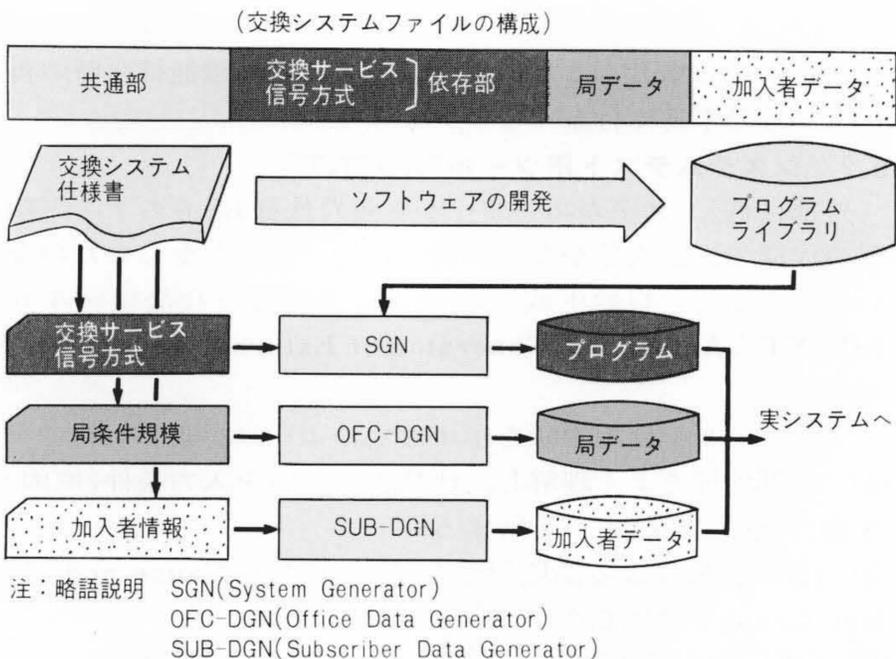


図6 FGNの構成 交換システムファイルの構成、オンラインファイル作成手順及び各種ツールの関係を示す。

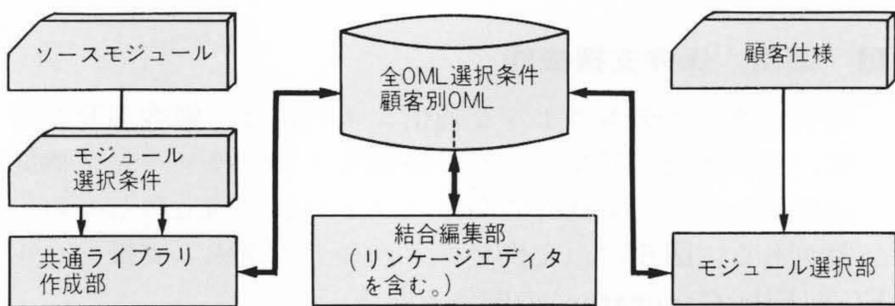


図7 SGNの構成 顧客仕様を会話形式で入力することによって、オンラインシステムを作成するツールの構成を示す。

一方、本ツールは、生成したシステムごとのファイル情報管理機能により、システム納入後の保守作業の効率化に対しても効果がある。

6.2 オフィス用データゼネレータ“DGN”

顧客の納入先によって、局条件、加入者条件などのシステムの規模面からみた要求が異なるため、図6に示すようにOFC-DGN (Office Data Generator), SUB-DGN (Subscriber Data

Generator)などのデータトランスレータを用意し、オンラインシステムが参照可能なLM(Load Module)形式のデータファイルを作成し、前述のSGNの出力ファイルと結合してオンラインファイルを作成する。

7 結 言

ソフトウェア生産技術の高度化として、各種通信システム用ソフトウェア開発支援システムの概要と代表的な支援ツールの機能について述べた。

本システムは、HITAC MシリーズのTSS環境化で、マイクロコンピュータを利用した各種通信システムの開発に使用しており、所期のねらいとした効果を達成している。

更に、各開発フェイズを横断的にみた生産管理用ツールMSSP(Management System for Software Production)は、急増するソフトウェア生産活動及び各種ソフトウェア生産技術の客観的評価に威力を発揮している。

今後、更に本システムの適用範囲の拡大、運用面からの課題のフィードバック及び機能拡充を推進し、高性能化を図っていく考えである。

最後に、本システムの開発及び導入に当たり、御協力をいただいた日立通信システム株式会社、日立ソフトウェアエンジニアリング株式会社の関係各位に対し、感謝の意を表わす次第である。

参考文献

- 1) 黒崎, 外: 電子交換機用仕様記述言語“SDL”, 日立評論, 62, 12, 915~918 (昭55-12)
- 2) 二村, 外: プログラムの木構造化図面“PAD”, 日立評論, 62, 12, 871~874 (昭55-12)
- 3) 前沢, 外: 処理論理図自動作成システムの開発, 情報処理学会, 第23回全国大会 (昭56-10)
- 4) 神野, 外: 16ビットマイクロコンピュータ 68000用高級言語S-PL/Hの開発, 情報処理学会, マイクロコンピュータ研究会, 17-3 (昭56-7)
- 5) 古川, 外: 機能テストのためのテスト項目作成手法について, 情報処理学会, ソフトウェア工学研究会 (昭55-11)
- 6) 桧山, 外: 分散形状態遷移方式における交換ソフトウェアの一方式, 電子通信学会, 交換研究会 (昭56-12)