

## モジュール・インターフェース 設計支援システム

編田 政之、石原 平太郎、青柳 廣、  
沖電気工業株式会社

吉田 徹、常田 高志  
沖通信システム株式会社

### 1. はじめに

大規模なソフトウェアの開発においては、構造的なブレークダウンを効率良く行い、モジュラリティーの優れたシステムを構築する事が極めて重要である。

交換ソフトウェアは電子交換システムの制御を司る大規模なソフトウェアシステムであり、極めて厳しい信頼性と高い保守性を要求されている事から、モジュールの分割技術が重要な技術の一つとなっている。この傾向は、近年の複合通信の発達により交換、通信ソフトウェアが複雑化してきていることから、急速に顕著になってきている。そこで、設計支援環境としてモジュール分割をサポートすることが重要な要素となっている。

今回、交換ソフトウェア用設計支援システムの一環としてこの領域をサポートするシステムを開発中であるので、その概要について以下に報告する。

以下では、「モジュール」は「機能ブロック」と同じ意味に用いる。

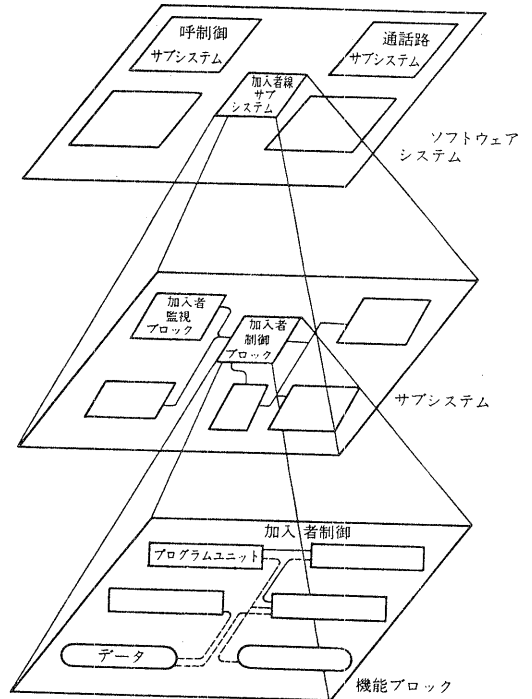


図1 ソフトウェア構造

### 2. ソフトウェア設計過程

#### (1) 階層化設計

交換ソフトウェアは、図-1に示す様にシステム-サブシステム-機能ブロックといった階層で構成されている。各階層に対応した設計ドキュメントを表-1に示す。

機能ブロックは交換機を外部からみたときの「機能」に着目してサブシステムを分割したものであり、CHILL (C C I T T プログラミング言語) 上の「MODULE (モジュール)」と対応している。機能ブロックの分割例を図-2に示す。

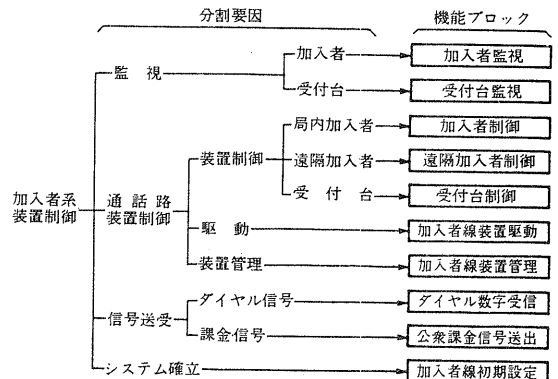


図2 機能ブロック分割例

(2) モジュールの構造と

モジュール間インタフェース

モジュールは複数のプログラムユニットと複数のプログラムデータにより構成する。

モジュール相互は、予め決められたプログラムユニットに対してのみ結合可能であり、これをモジュールインタフェースと呼ぶ。モジュール内はプログラムユニット相互で結合可能であり、これをユニットインタフェースと呼ぶ。また、各プログラムユニットは、モジュールデータに対するアクセスは可能であるが、他のプログラムユニットのデータ（ユニットデータ）にはアクセスできない。

モジュールの構造を図3に示す。

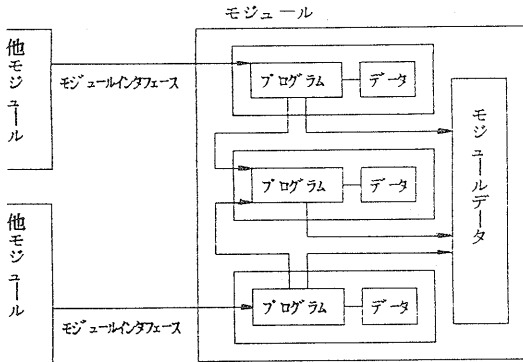


図3 モジュールの構造

区分	ドキュメント名称	略称
D S	ブロックインタフェース仕様	BI
	ブロックインタフェースパラメータ一覧	BPL
	ブロックインタフェースモード定義一覧	BML
	ブロックインタフェースモード構成	BMC
	プロセッサ間インタフェース仕様	PI
	コントロールインタフェース仕様	CI
	ブロックデータ構成図 局データ構成図	BD OD
DDS	ユニットインタフェース仕様	UI
	ユニットインタフェースパラメータ一覧	UPL
	ユニットインタフェースモード定義一覧	UML
	ユニットインタフェースモード構成	UMC
	ユニットデータ構成図	UD

表1 設計ドキュメント一覧

3. 設計支援環境としての要求条件

ソフトウェア設計工程での機械的作業の大部分は、設計ドキュメント作成に占められる。特に、交換ソフトウェアの設計ドキュメントは膨大であり（1万頁を越える）、そのため設計ドキュメントを効率よく作成することが重要となる。

3.1 現状の問題点

設計工程では、次のような問題がある。

(1) 設計ドキュメントの作成・修正に多大な工数を要す

プログラム規模が大きいこと、階層的ドキュメント構成をとっているために重複した記述があること等により設計ドキュメントの記述量が多い。近年、ワードプロセッサで作成しているが、ドキュメントの作成・修正に多くの工数を要しており効率の良い専用エディタが望まれる。

(2) 既存の設計情報 (ドキュメント) の再利用が難しい

手書きドキュメントはもちろんのこと、汎用のワードプロセッサでもドキュメント相互間の関連がないこと、必要な設計情報の検索機能がないこと等により、既存の設計情報の再利用が難しい。

(3) レビューに工数がかかる

ドキュメント相互間の機械的チェックも人手で行っている。

(4) 設計ドキュメントとプログラム間に不一致が生じやすい

プログラム製造時、設計ドキュメントを見ながら人手でコーディングするため、誤りを生じやすい。

また、デバッグ終了時、プログラムの修正に伴い設計ドキュメントを修正することが大変である。

以上の背景に基づいて、設計支援システムの開発を進めている。

### 3.2 設計支援システムの必要機能

(1) 設計業務のEDP化 (清書機能, 等)

(2) 他システムの設計情報の再利用 (設計情報のデータベース化)

(3) レビュー支援 (設計情報の一貫性チェック等)

(4) 保守用ドキュメントの自動生成

(5) 設計情報よりプログラム自動生成

(6) プログラムより設計情報作成

### 3.3 設計支援システムの構成

設計支援システムの構成を図4に示す。

設計情報データベースは、交換ソフトウェアのプログラム構造に対応してシステム—サブシステム—機能モジュール—プログラムユニットという階層を持ち、また各種ドキュメントの書式およびプログラム生成時等に使用する翻訳辞書をもつ。

エディタは、ドキュメント体系に対応した会話型専用エディタであり、設計情報のデータベースへの格納、修正、検索を行なう。また、他システムの設計情報のコピー、即ち設計情報の再利用を行なう。

レビュー支援は、設計情報階層間の一貫性をチェックする。

ドキュメント作成は、指定された書式に従い設計情報を高速プリンタを使用して、ドキュメントを出力する。設計支援システムは、製造支援システムでのプログラム生成およびプログラム解析とインタフェースをもち、ソースプログラムの自動生成およびソースプログラムから設計情報の抽出を可能とする。

モジュール間インタフェースの設計は、エディタ部とレビュー支援部によって実現されている。

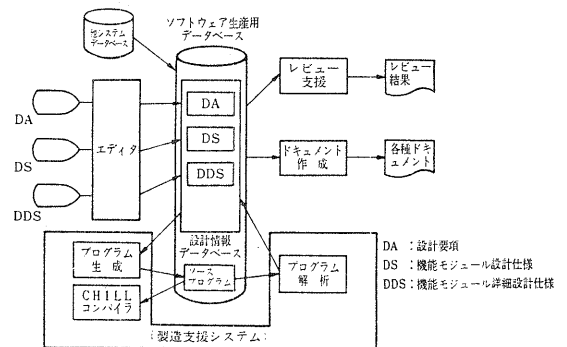


図4 設計支援システムの構成

#### 4. モジュールインタフェース

##### 設計支援システム

設計工程における支援環境として、ソフトウェアの部品化による再利用システム等、最近様々な方式が実現されているが、我々は設計者自身が作業形態について案外保守的であると考え、まず最初の支援環境としては設計者が作るドキュメントに対する支援ということにした。

即ち、交換ソフトウェアのドキュメントがプログラムのモジュール構造に合致した体系になっていることに着目し、そのドキュメント作成を支援すると共に、設計時のモジュール間インタフェースの整合性をチェックする支援システムを作成することにした。

設計作業を、「システムをトップダウンで詳細化したモジュールに分割していき、そのモジュールでは機能とモジュール間のインタフェースを明確にしていく作業である。」と定義した。さらにインタフェースとは、モジュールとモジュールとの接点の仕様であり

- ① プログラムユニット名称
  - ② プログラムユニットの属性
  - ③ パラメータの属性
  - ④ パラメータの個数
- とした。

#### 4. 1 主な機能

##### (1) 設計情報編集機能

TSS 端末を設計者のワークシートの代替として使用できるように設計情報の編集機能を盛りこんだ。編集機能には、次に示すような機能を用意した。

- ① モジュール構造の各レベルにおける複写機能

この機能により既存システムからの設計レベルでのプログラム構成を含む流用を可能にした。

- ② モジュール構造に対する追加，変更，削除，登録の機能

- ③ インタフェースの各項目に対する追加，変更，削除，登録の機能

ブロックインタフェースパラメータを登録する画面例を図5に示す。

##### (2) 用語定義機能

ニーマニックで使用される用語に対する日本語の定義機能であり、設計レベルからプログラムレベルまでの用語の統一を目的とする。

用語辞書には全システム共通辞書と個別システム毎の辞書を持ち、変換処理の効率を図っている。

辞書による変換例を図6に示す。

OTS	ブロックインタフェースパラメータ一覧	(BPL)
1	パラメータ，構成要素名： TKSS 意味： 中継線サブシステム 使用インタフェース名： モード．配列1次下限：            上限： レベル： 1モード： BIN レンジ下限：	セット要素値：  10： -  2次下限：            上限： 長さ： 上限：
2	定義： パラメータ，構成要素名： 意味： 使用インタフェース名： モード．配列1次下限：            上限： レベル：            モード： レンジ下限：	セット要素値：  10：  2次下限：            上限： 長さ： 上限：

図5 BPL登録画面例

(3) ドキュメント出力機能  
高速漢字プリンタに所定の書式に編集し、設計書を印刷する。

出力するドキュメント一覧を表1に示す。

(4) モジュール間のインタフェースのチェック機能

モジュール間のプログラムユニットとパラメータの参照関係を内部テーブル上でチェックし、差分を出力する。

#### 4.2 システム構成

図7に全体のシステム構成を示す。居室に設定されたTSS端末、モジュールインタフェース等の設計情報を蓄積するための大容量ディスク及びペーパードキュメントを出力するための高速漢字プリンタを特徴とする。

エディタ部は、次のモジュールにより構成される。

(1) システム制御モジュール

(2) システム定義モジュール

(3) システム変更モジュール

(4) 設計情報編集モジュール

(設計情報エディタ)

(5) 複写処理モジュール

(6) 削除処理モジュール

モジュール構造とインタフェースを内部表現するためにモジュールの包含関係を表したトリートリー形式のモジュール管理テーブル、プログラムユニット属性テーブル、パラメータ属性テーブル等をディスクファイル上に作成した。テーブルの構成は将来プログラム設計情報を追加できるよう考慮した構成となっている。エディタの機能を実現するのに、そのトリートリーのリンクのハンドリングとテーブルデータの変更により編集機能を実現している。

#### 辞書による変換例

MN を入力する。

日本語変換： MN

辞書検索キー を押す。

日本語変換： 多重度

再度、辞書検索キーを押す。

日本語変換： マイナー

AL を入力する。

日本語変換： マイナーAL

辞書検索キーを押す。

日本語変換： マイナーアラーム出力

図6 辞書による変換例

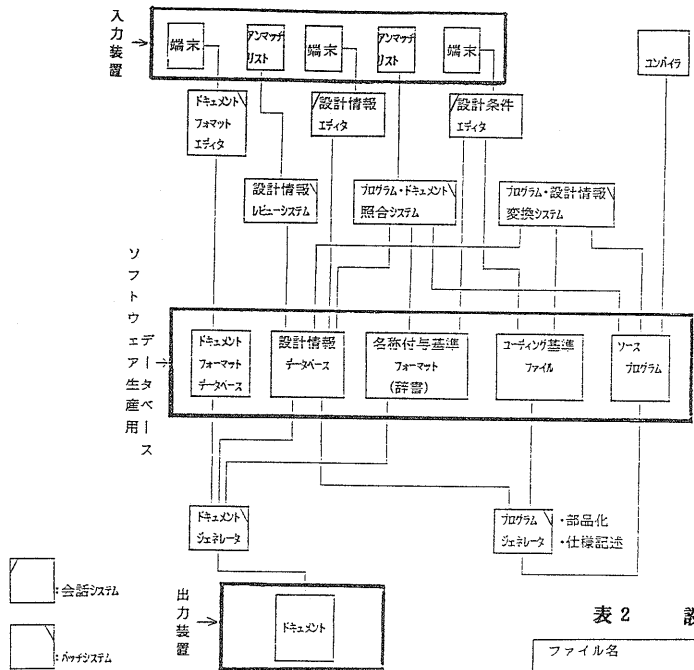


図7 システム構成

表2 設計情報データベース・ファイル構成

ファイル名	機能概要
設計情報管理ファイル	本データベースをシステム単位に管理するための情報を持つ。 設計支援システムに1ファイル存在する。
設計情報ファイル	登録されるシステムに対して1ファイル作成され、設計情報そのものをもつ。 注 参照
用語辞書ファイル	設計書等で用いられる略語・用語・データ名称の意味を定義したファイル。

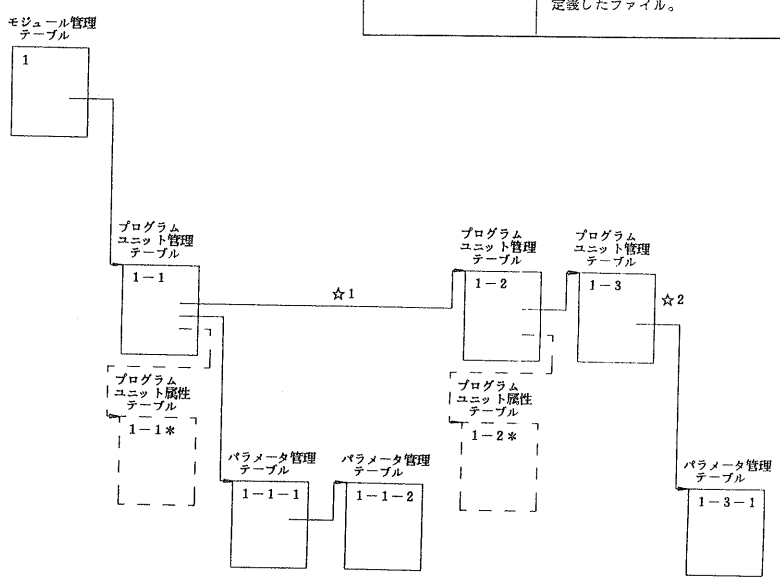


図8 設計情報データベース・テーブル構成

## 5. あとがき

以上に述べたシステムは、現在、開発中であり完成後設計支援システムとして実際の交換ソフトウェアの開発に使用される予定である。

今後の予定として、HCPチャートプロセッサ(2)との統合が計画されている。

HCPチャートは、処理の階層的に記述に非常に適した性質を持っており、以上に述べたモジュールの階層的設計手法と親和性がある。これにより、モジュールインタフェース設計から処理の設計、コーディングまで一貫してサポートする支援環境が実現される。

## 6. 参考文献

- (1) 高橋 他：ソフトウェア生産技術の現状と将来特集，電子通信学会，電子通信学会誌，Vol. 66，No. 4，p. p. 321～419，1983
- (2) 内田，青柳，田中：HCPチャートプロセッサ，情報処理学会，第28回全国大会論文集，p. p. 529～530，1984
- (3) 内田，石原，青柳：交換プログラム用生産システム統合化の検討，電子通信学会，昭和59年度全国大会1，p. 431，1984
- (4) 西尾：交換用テーブル設計・製造ドキュメント作成支援システム的一种方式，電子通信学会，昭和59年度全国大会，p. 354，1984
- (5) 常田，石原，青柳，鍋田：モジュール・インタフェース設計支援システム，情報処理学会，第30回全国大会，6T-8

(6) 森口，武石：D70自動交換機のソフトウェア構成，沖電気研究開発121Vol.50 No.3

(7) 内田，石原，青柳：電子交換用ソフトウェアの開発環境，沖電気研究開発125Vol.52 no.1