ソフトウェア工学 41－2
（1985. 5. 27）

設計情報管理ツールを利用したシステム開発

富士通（株）
吉村　博史

はじめに

信頼性が高く，拡張等の保守が容易なシステムをいかに効率良く作るかということはシステム開発における大きな課題である。

一般のソフトウェア工学除を産業の生産性・品質の向上を図るためには，プログラミング・テスト・ツール等の開発，ソフトウェア工程管理ツールの開発，ソフトウェア工程

1．設計情報管理ツールの開発背景

設計情報開発ツールが開発された四つの背景について述べる。

1－1　データ項目管理の必要性

データベースを用いたシステム開発では，データベースの設計・維持の際に，データの統合化・共有化を図るためには，データの変更等の影響が広範囲にわたるので，その影響を的確に把握するためには，データベースの設計段階からデータ項目を管理する必要がある。

1－2　ディクショナリ情報の一意化の必要性

保守作業のために，ディクショナリから相互関連情報を検索して影響調査を行う場合，ディクショナリに登録されている情報について，同一実体が一意に管理されないと，調査結果に過不足が発生する。

ソースプログラムを入力するディクショナリ利用者が，同一システムの場合，全てのソースプログラムで一意に使用されていなければ，ディクショナリの効果を発揮できない。

プログラムが完成した後で，データ項目情報を一意にしようとしても，ソースプログラムの修正作業が多く，業務に支障をきたす。

これを防ぐには，ディクショナリに登録する情報を設計工程から一意に決定しておく必要がある。

1－3　設計情報の正しい伝達の必要性

システム開発では通常，設計と製造の分離が行われる。特に大規模なシステム開発では，多くの人々の間で情報の伝達を正しくスムーズに行うことが重要である。

そのため，各サブシステム間で機能やデータ項目等の名称に間違いがないようにし，それらを統一することで，設計書の誤解による正しい伝達の発生を減少させることができる。

1－4　データ関連の品質向上の重要性

生産性・品質の向上を実現した，過去のプロジェクト報告［1］で分析された，システムテスト以降でのエラーの内容と原因工程の割合を表1に示す。

表1．システムテスト以降でのエラーの分析

原因工程 / 内容	～SD	PS	MD	PG	合計	
ロジック	3.1	27.5	3.4	4.5	38.5	
データ操作	2.3	12.4	2.1	7.9	24.7	54.3
データ定義	3.2	7.2	0.3	7.2	17.9	
インタフェース	7.9	3.8	0	0	11.7	
入出力	1.0	1.0	0	0	2.0	
その他	4.2	1.0	0	0	5.2	
合計	21.7	52.9	5.8	19.6		
	74.6				100	

SD：システム設計
PS：プログラム構造設計
MD：モジュール設計
PG：プログラミング

この表より，システムテスト以降に発生したエラーのうち，74.6%がプログラム構造設計以前に作り込まれたものであり，54.3%が「データ操作・データ定義・インタフェース」等のデータ関連のエラーであることがわかる。これから，プログラム構造設計工程以前でのデータに関する品質向上が，システムの最終的な品質向上にかなり影響していることが推測される。

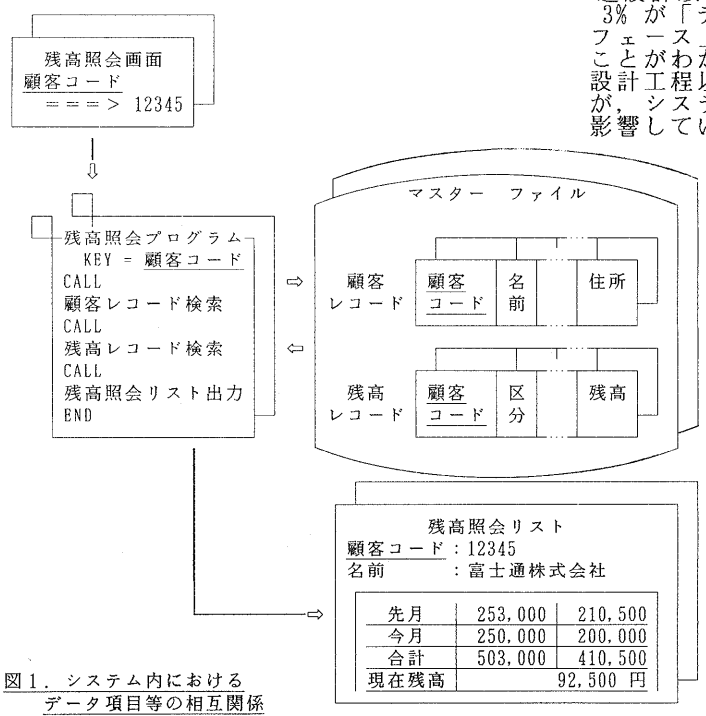


図1．システム内における
データ項目等の相互関係

図1にシステム開発におけるプログラム等の相互関係を示す。下線部分のプログラム内で共通項目のプログラム・レコード・データ・ファイルが使用されていることがわかる。

前述したように，4つの背景で示した産業界のシステム開発でのデータ項目管理，品質向上，工程長短縮，属性の統一，レコード・データ名称の統一，設計情報の正しい伝達，データ関連の品質向上などは，ソフトウェア工程管理の中心となるプログラム等の相互関係を早いうちに明確にすることが必要である。

多量で複雑に関連しあう情報を設計工程から管理するには，これらの手段を作成した。

2．設計情報管理ツールの開発目的

前述の背景のもとで，設計段階から生産性・品質の高いシステムの開発をサポートするために，以下の目的で開発を行った。

2－1　設計情報の一元管理

データ項目名・レコード名・ファイル名モジュール名（機能の名前）等の一意性の重要性は前述したが，修正の時間的なズレや修正もれによる管理情報間や出力情報間の不整合を無くすため，一元管理を行う。

2－2　設計の品質向上

管理情報の相互関係の矛盾チェックを行い，チェックもれの防止とチェック時間の短縮をサポートする。
また，異なるレコードで使用されるデータ項目の属性を等しくすることにより設計工程からデータ定義の品質向上を図り，以降の工程におけるデータ関連のエラーの減少を図る。

2－3　修正作業の省力化

修正時の影響範囲を相互参照により明確にし，設計の変更・追加発生時や，及び設計ミス修正時の影響調査の時間を削減し，修正もれをなくせるようにする。

2－4　設計ドキュメント修正の負荷軽減

ファイル設計・プログラム間インタフェースを中心とした設計ドキュメントの出力をサポートする。
これにより，最新ドキュメントの保持のために必要であるドキュメント修正作業の負荷軽減をはかる。また一元管理された設計情報によりドキュメントを出力することにより，修正もれ・ドキュメント間のずれを無くす。

2－5　設計情報を次工程へ正しく伝達

一元管理された設計情報と整合性が保証されたコピーライブラリ等を自動生成する。
これにより，プログラミング工程でのデータ定義に関する製造工程を大幅に短縮し，さらに品質も保証する。また，設計ドキュメントと自動生成したコピーライブラリ間のずれを無くしドキュメントの信頼性を高める。

3．設計情報管理ツールの機能と特長

3－1　情報の登録・更新・削除・参照

＊対象　以下の四つの単位で，設計工程より情報の登録・更新・削除・詳細検索・一覧検索をサポートする。
表2に，これら四つの各管理情報の詳細を示す。
①データ項目情報
②ファイル情報
③レイアウト情報
④モジュール情報

表2．設計情報管理ツールでの管理情報

NO	データ項目情報	ファイル情報	レイアウト情報	モジュール情報
1	データ項目名（英数字）	ファイル名（英数字）	コピー句名（英数字）	モジュール名（英数字）
2	データ項目名（日本語）	ファイル名（日本語）	データ項目名（英数字）	モジュール名（日本語）、
3	データ項目名（フリガナ）	ファイル名（フリガナ）	データ項目の階層（レベル）	モジュール名（フリガナ）
4	数字・英数字等のタイプ	DB,DS,TBL の種別	基本・集団項目の区分	COBOL,ASM の言語区分
5	桁数	PS,PO etc のファイル編成	PACK,ZONE etc のタイプ	2次入口点名（英数字）
6	小数点桁数	F,V,FB etcのレコード形式	繰り返し回数	2次入口点名（日本語）
7	符号	ブロック長	再定義の項目名（英数字）	2次入口点名（フリガナ）
8	単位	レコード名（英数字）	その他詳細な情報	副モジュール名（英数字）
9	取り得る値（値域）	レコード名（日本語）	レコード長	ファイル名（英数字）
10	データ項目の意味・内容	レコード名（フリガナ）	作成年月日	レコード名（英数字）
11	作成年月日	コピー句名（英数字）	更新年月日	コピー句名（英数字）
12	更新年月日	コピー句名（日本語）	更新回数	メイン主プロ名（英数字）
13	更新前の内容	コピー句名（フリガナ）		作成年月日
14		作成年月日		更新年月日
15		更新年月日		

＊処理方式
①バッチ処理による一括登録.
②バッチ処理による一括更新.
（レイアウト情報のみ）
③日本語メニュー方式による，対話登録・更新・削除・詳細検索・一覧検索.
＊特長
①日本語入力方式として，単語単位のカナ漢字変換（ホストで変換）・日本語直接入力等が可能.

３－２　各種チェック機能

＊対象と処理方式
①相互関連チェック…日本語メニュー画面からのバッチジョブ起動による，６種類の一括チェック.（表３）
②登録・更新時の，未定義チェック・重複チェック・文法チェック.
＊特長
①相互関連チェックの場合，画面から範囲指定ができる.
②重複チェックは，外付けで範囲指定ができる.

３－３　生産物の自動出力

＊対象と処理方式
①ドキュメント…日本語メニュー画面からのバッチジョブ起動による，１１種類の設計ドキュメント出力.（表４）
②次工程で使用する以下のライブラリの自動作成.
・ＣＯＢＯＬプログラムのデータ定義域で使用するコピーライブラリ.
・アセンブラプログラムのデータ定義域で使用するマクロライブラリ.
・ソースプログラムを入力として保守用日本語ドキュメントを出力するツール（ＳＮＯＴＥ：Standard NOTation Editing system ）で使用する日本語名標辞書.
＊特長
①ドキュメントの出力時，画面から範囲指定ができる.

表３．相互関連チェックの種類とチェック内容

NO	相互関連チェック	チェック内容
1	レコード　⇨ データ項目	レイアウト情報で定義しているデータ項目の未定義チェック・属性不一致チェック・レイアウト内重複チェック
2	モジュール　⇨ レコード	モジュール情報で定義しているレコードの未定義チェック
3	モジュール　⇨ ファイル	モジュール情報で定義しているファイルの未定義チェック
4	モジュール　⇨ 副モジュール	モジュール情報で定義している副モジュールの未定義チェック
5	設計情報　⇨ ソースモジュール	モジュール情報の管理内容と実際に作成されたソースプログラムとの矛盾チェック
6	USER-EXIT 属性チェック	レイアウト情報で定義されたデータ項目のタイプ毎の標準化チェック （チェック項目はユーザ指定可能）

３－４　相互関連情報管理

＊対象と処理方式
①相互関連情報検索…日本語メニュー画面からのバッチジョブ起動による１０種類の相互関連情報検索.
（表５）

表５．相互関連情報

	相互関連情報		相互関連情報検索の内容
1	データ項目　⇨レコード	指定された	データ項目を使用しているレコードの一覧
2	レコード　⇨ファイル		レコードを使用しているファイルの一覧
3	データ項目　⇨モジュール		データ項目を使用しているモジュール一覧
4	コピー句　⇨モジュール		コピー句を使用しているモジュールの一覧
5	レコード　⇨モジュール		レコードを使用しているモジュールの一覧
6	ファイル　⇨モジュール		ファイルを使用しているモジュールの一覧
7	副モジュール⇨モジュール		副モジュールを使用しているモジュール一覧
8	二次入口点　⇨モジュール		二次入口点を定義しているモジュール一覧
9	コピー句　⇨レコード		コピー句を使用しているレコードの一覧
A	モジュール関連情報 上位モジュール 下位モジュール		モジュールを直接・間接的に呼び出しているモジュールの一覧，又は，指定モジュールが直接・間接的に呼びだしているモジュールの一覧

②指定されたデータ項目を使用しているレイアウト情報の一括更新.
＊特長
①相互関連情報検索は，画面から範囲指定ができる.

以上の機能を実現するための，設計情報管理ツールのシステム構成を，図２に示す。

表４．出力ドキュメントとその内容

NO	ドキュメント	ドキュメントの内容	
1	データ項目台帳	データ項目の一覧	
2	ファイル一覧	ファイル情報の一覧	
3	レコード一覧	レコード名の一覧	
4	データ領域定義	レイアウト情報詳細	
5	レコードレイアウト	レイアウト図	
6	変更項目対比一覧	NO 4の変更箇所一覧	
7	モジュール詳細	モジュール情報詳細	
8	モジュール一覧	モジュール名の一覧	
9	プログラム構成図	親子関係	図式目次表示
10	プログラム関連図	間接関係	マトリックス
11	プログラム構造図	モジュール関係	構造図

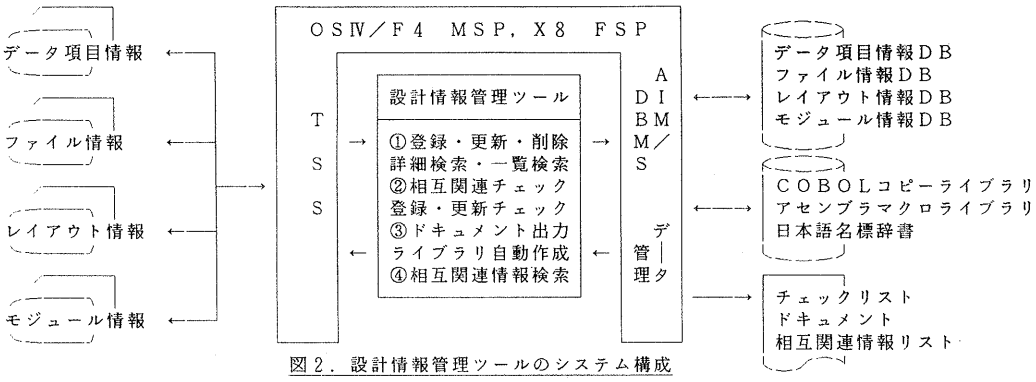


図２．設計情報管理ツールのシステム構成

（例）データ領域定義

	設計書番号	業務名	項目番号	標題	データ領域定義	照査者
				(／)		

種別	□ DB □ テーブル ■ DS □	ファイル名 (N) (A/N) TESTFIL1	レコード名 (N) テストレコードＹＹＹ (A/N) XXXX1YYY	ＣＯＰＹ句名 XXXX1YYY	媒体	ファイル編成 順編成

レベル番号	データ項目名 (N)	データ項目名 (A／N)	属性１（P）	属性２（M）	バイト数	相対位置	繰返し回数
03	カキケ	YYYDDTESTAA1				1	
05	テストデータＡ５	YYYDDTESTA5	9(1)		1	1	3
05	テストデータＡ１	YYYDDTESTA1	9(3)		3	4	
05	テストデータＡ４	YYYDDTESTA4	9(3)		3	7	2
05	テストデータＮ３	YYYDDTESTN3	1(4)	BIT	1	13	
05	テストデータＮ２	YYYDDTESTN2	1(5)	BIT	1	13	
03		YYYDDTESTBB				15	
05	テストデータＢ１	YYYDDTESTB1	9(4)		4	15	
05	テストデータＢ２	YYYDDTESTB2	9(4)		4	19	
05	テストデータＢ３	YYYDDTESTB3	9(4)		4	23	
03	テストデータＣＣ	YYYDDTESTCC	S9(6)		6	27	

（例）データ項目⇨レコード

---------- 相互関連情報 （データ項目 ---> レコード） ----------

走査条件 ： YQDATA01

データ項目名（英数字） ： YQDATA01

データ項目名（日本語） ： テストデータ００１

属　　性 ： SV9(1)

項番	レコード名（英数字）	レコード名（日本語）	コピー句名	レコード長	
00001	LAYOUT61	テストレコードＡＡＡＡＡＡＡ	LAYOUT61	00384	SV9(1)
00002	YQRECA00	富士通株式会社レコード	YQRECA00	00006	SV9(1)
00003	YQ1CPY77	情報処理学会レコード	YQ1CPY77	00001	SV9(1)
00004	YQ1REC01		YQ1REC01	00001	SV9(1)
00005	YQ5REC98	レコードデータ１	YQ5REC98	00001	SV9(1)

4．設計情報管理ツールの適用

4－1　設計作業と設計情報登録との関係

設計情報の洗い出しは，工程毎の設計手順に準じて行われる。

図3に，当社のＳＤＥＭ（Software Development Engineering Methodolgy）のソフトウェア開発作業標準に基づいた開発工程に従い，設計作業と設計情報管理ツール適用のための設計情報の登録・更新作業の関係をプロセスフローで示す。

①データ項目情報の登録

初期設計にシステムや事務処理に関する基本ドキュメントを用いて，基本的な事務処理用語等のデータ項目を洗い出す。この中から属性等を決定した後，データ項目名，項目情報を登録する。

②ファイル・レイアウト情報の登録

①で洗い出されたデータ項目情報をもとに，ファイル設計工程において，データ項目の関連構造を検討する。当ファイル・レイアウト情報の登録・更新作業は，データ項目情報の登録と同時に行う。

ファイル設計等によりデータベースやファイルを決定し，データ領域，レイアウト情報の登録する。この時，データ項目情報とファイル・レイアウト情報とを関連付けて管理されているものでなければならない。

③モジュール情報の登録

ファイル設計のフェーズが終了した時点で，プログラム構造設計や機能分割を行おう。モジュールの親子関係を領域間の受け渡しの情報と，モジュール間のインターフェイスを決定する。

プログラム構造設計は，モジュール情報の登録・更新作業と，モジュール情報の登録・更新作業は，モジュール情報の登録・更新作業は，モジュール情報の登録・更新作業は

5．設計情報管理ツールの効果

設計情報管理ツールの機能を生かして，当初の目的を達成するには，設計情報の維持を正しく行っていかなければならない。ここでは，大規模オンラインデータベースシステム開発での当ツールの適用状況をもとに，当ツールの評価を行う。

5－1　効果

①設計情報の管理をツールで行うことの重要性と効果を設計者が認識した。
②標準化等の規約が単なる標準に終らずに，チェックされることにより徹底された。特に，当ツールでチェックされた命名規約については十分に標準化された。
③当ツールの出力ドキュメントの「データ領域定義」は，正式ドキュメントとして，各設計チームやソフトウェアハウスに，同時に同じ内容のものが提供されるため，作業担当者間の誤解が防げた。
④データ項目のタイプ，桁数などが修正になった場合等，それが複数のレコードで使われていても，全てのレコードの修正が保証されるため，データ定義に関する信頼性が向上した。

5－2　問題点

①管理情報の維持作業の負荷が，設計者運用者ともに高い。
②設計情報は，設計が終了しても次の設計で使用されるため，その世代管理が必要である。
③モジュール情報については，その情報からソースプログラムを作成していない為ソースプログラムと情報の乖離が起き易い。

6．設計情報管理ツールの今後の課題

図4に設計情報管理ツールが今後進むべき5つの方向を示す。太線で囲ってある枠がその内容である。

実際の適用において発生した問題や，今後発生するであろう新たな要件も，全てこの中に吸収されると考えている。

図5に，既存開発支援ツールと設計情報管理ツールとの関連を工程を横軸にして示す。これらの既存開発支援ツールとの整合性をとりながら，より高いシステム開発の生産性・品質向上の向けてレベルアップすることが，今後の課題である。

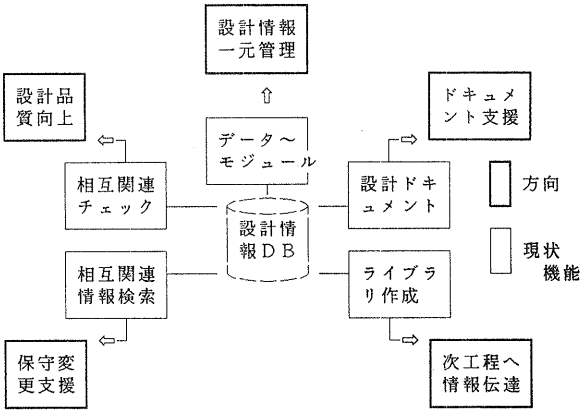
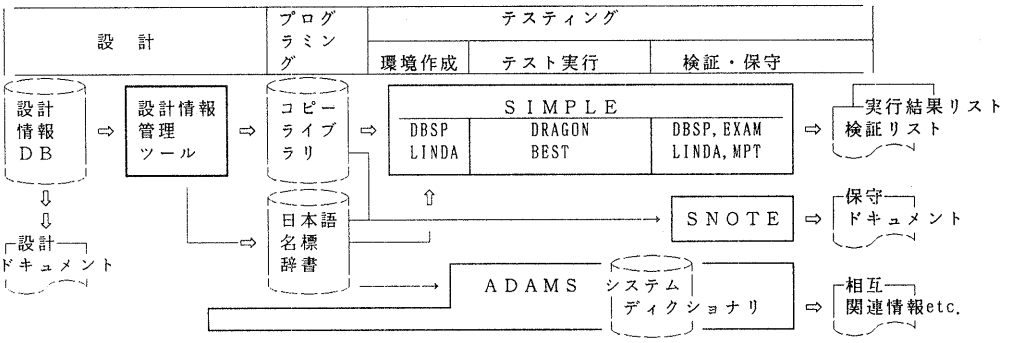


図4．設計情報管理ツールの今後の方向

－参考文献－

［1］　八田：”アプリケーションプログラム開発での品質管理事例”
日科技連第3回ソフトウェア生産における品質管理シンポジウム発表報文集 PP.79-86,(1983)



ＳＩＭＰＬＥ	:	SIMPLE application development and management system
DBSP	:	Data Base Support Program
LINDA	:	Logical INformation support tool of DAtaset
BEST	:	Batch program Execution Support Tool
DRAGON	:	DRiver And stub Generator for ON-line and batch test
EXAM	:	EXecution stAtics Measurer
MPT	:	Module Path Tracer
ＡＤＡＭＳ	:	Application Development And Management System

図5．工程毎の開発支援ツールの関連