

ソフトウェアの「日本語による仕様書」 作成支援システム

大石 東作
(電子技術総合研究所)

ソフトウェアの作成においては、プログラム自体の作成のみならず、各種の仕様書の効率の良い作成が、プログラムの機能確認・保守・機能改良に不可欠である。しかしながら、従来は仕様書の作成や管理は、完全に人手で実施されており、計算機システムからの支援を受けることは出来なかった。

本システムは、「日本語による仕様書」の効率的な作成を支援するものであり、人手により記述された仕様書の部分や、仕様を付加させたプログラムから、プログラムと仕様を統合したものを作成する。これを、DocuGram (仕様書+プログラム) と呼ぶ。また、必要に応じてDocuGramは、整形化した単体の仕様書やプログラムへと自動的に合成・管理される。

A Support System for Software Documents in Japanese

Toskau OISHI

Information Systems Section, Computer Science Division

ELECTROTECHNICAL LABORATORY

1-1-4 Umezono, Tsukuba Science City, Ibaraki 305, Japan

In software production process, it is important to make some software documents efficiently, but effective supports by computers for the documents generation are not available now, especially in Japanese environment.

A support system for software documents assists efficient documents generation. Specifications of the software in Japanese and program codes are written manually, then they are integrated to DocuGram : (Document + Program). The docuGram is translated automatically to some documents, such as the outline specification, design specification and pure program codes by a formatter.

1、はじめに ソフトウェアの作成においては、プログラム自体の作成のみならず、各種の仕様書の効率の良い作成が、プログラムの機能確認・保守・機能改良に不可欠である。しかしながら、従来からの仕様書の作成や管理は、ほとんど人手で実施されており、計算機システムからの支援といっても、日本語ワードプロセッサを使用する程度であった。

本システムは、「日本語による仕様書」の効率的な作成を支援するものである。人手により記述された仕様書の部分や、仕様を付加させたプログラムから、プログラムと仕様を統合したものを作成する。これを、DocuGram（仕様書+プログラム）と呼び、保存する。また、そのときの必要に応じてDocuGramは、整型化した単体の仕様書やプログラムへと自動的に変換・合成・管理される。

2、仕様、プログラム、各種の仕様書 ソフトウェアの生産性を質的に向上させる方式として、「実行可能仕様にもとづく変換プログラミング」[1]に期待がよせられている。実行可能仕様においては、システム作成者と計算機のインタフェースとしての具体性が重視される。しかし、現実のプログラムの仕様としては、システム発注者あるいはユーザとシステム作成者のインタフェースとしての抽象性に重点が置かれている。なぜなら、システム発注者やユーザは、システムの機能や種々の制約条件を大まかに捉えることが可能であることが、まず肝要だからである。そして、システムの具体性は、システム作成者と計算機のインタフェースである手続き（プログラム）によって実現される。ところで、抽象的な仕様と、具体的な機能を実現する手続き（プログラム）の間には、大きな意味的なギャップがあり、前者から効率の良いプログラムを機械的に導出するのは困難である。

現在、システム発注者あるいはユーザとシステム作成者のインタフェースとしての仕様は、自然言語で記述されるのが普通である。当然我が国では、日本語が使用される。

上記の仕様は、各種の仕様書として一般的に流通している。仕様書の種類は、時と場合に応じて様々なものがあり

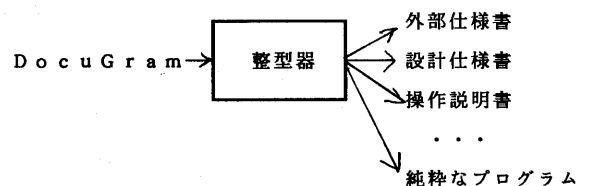
えるが、ここでは次のものを取りあげることにする。

- ・ 外部仕様書
- ・ 設計仕様書
- ・ 操作説明書
- ・ 検査仕様・成績書

これらの仕様書は、従来から日本語で記述されており、最近では日本語ワードプロセッサの助けを借りて作成されることが多くなっている。このことから、現在では日本語入力方法の進歩により、日本語による仕様書作成と英数字によるプログラム作成における、文字入力の手間の多寡のギャップがかなり小さくなっていると考えられる。これらのことは、仕様書作成とプログラム作成が、統合的に行える可能性を示唆していると思われる。

3、仕様書とプログラムの統合 プログラムとそれを説明するドキュメントを統合して利用することは、D. E. Knuth のWebシステム[2]等の例がある。

ここでは、各種の仕様書をプログラム中に、コメントの形式で統合する。この統合物を、DocuGramと呼ぶことにした。DocuGramは、第1図に示すように、整型器を通すことにより、各種の仕様書、純粋なプログラムへと変換され、ソフトウェアを管理する文書となる。



第1図、DocuGramの作成と整型化

DocuGramの1例として、前述の整型器自身を記述した、外部仕様書の部分を第2図に示す。それには、DocuGramの特徴等もしめされている。

第2図、DocuGramの例（整形器外部仕様書）

```

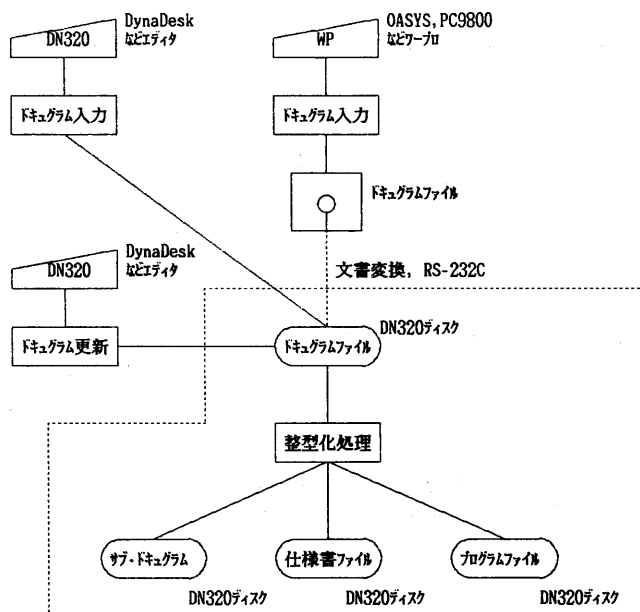
(*VB ===== *)
(*) 「日本語による仕様書」作成支援システム外部仕様書 (*)
(*)
(*) 1 概要 (*)
(*) 「日本語による仕様書」作成支援システムは、「日本語による仕様書」 (*)
(*) の効率的な作成を支援するものであり、仕様書の部分や、仕様を付加さ (*)
(*) せたプログラムから、プログラムと仕様書を統合したもの（ドキュグラ (*)
(*) ム）を合成する。 (*)
(*) また、必要に応じて、ドキュグラムから整形化した仕様書やプログラ (*)
(*) ムを生成する。 (*)
(*) 本システムから以下のような効果が得られる。 (*)
(*) ・仕様書作成の効率化が図れる。 (*)
(*) ・プログラムと仕様書の統合により、相互の矛盾を排除する。 (*)
(*) ・仕様書の一部をプログラムの静的定義部として有効利用すること (*)
(*) により、仕様定義の重複的記述を排除することが可能となり、プロ (*)
(*) グラム作成の効率化が図れる。 (*)
(*) ・プログラムの静的定義部の早期確定は、プログラムの機能仕様の (*)
(*) 人手による確認にも役立つ。 (*)
(*)
(*) 本プログラムは上記機能のうち整形化機能を実現する。 (*)
(*)
(*) 2 構成 (*)
(*) 本プログラムは、ワークステーションやパソコン上の日本語ワープロ (*)
(*) を用いて作成された「日本語による仕様書」の部分と仕様の付加された (*)
(*) プログラムの統合物（ドキュグラム）を入力して、整形化された「日本 (*)
(*) 語による仕様書」、プログラム、また必要に応じてドキュグラム（サブ (*)
(*) ドキュグラム）を出力する。 (*)
(*)
(*) 3 機能 (*)
(*) 先に作成されたドキュグラムを必要に応じて、整形化された「日本語 (*)
(*) による仕様書」ファイル、プログラムファイル、サブドキュグラムファ (*)
(*) イルへと変換する。 (*)
(*) 仕様書ファイルを印刷すれば通常のプログラム作成のための各種仕様 (*)
(*) 書が得られる。 (*)
(*) プログラムファイルは、通常のプログラム言語のコンパイラで処理さ (*)
(*) れる。 (*)
(*) サブドキュグラムファイルは別バージョンのドキュグラムファイルと (*)
(*) して本システムで処理できる。 (*)
(*) 整形化においては、仕様への整形化あるいはプログラムへの整形化の (*)
(*) 指定、整形化の程度の指定、対象となるプログラム言語の指定が行われ (*)
(*) る。 (*)
(*) なお、対象とするプログラム言語は、Pascal, C, Fortranとする。 (*)
(*)
(*) 4 出力物の種類と整形化程度 (*)
(*) ① 「日本語による仕様書」ファイル (*)
(*) I) 外部仕様書ファイル (*)
(*) II) 設計仕様書ファイル (*)
(*) ① 整形化の程度 (*)
(*) ・改頁／改行しない（指定なしの場合） (*)
(*) ・モジュール単位で出力（改頁） (*)
(*) ・手続き・関数・サブルーチン単位で出力（改頁） (*)
(*) ② 詳細化の程度 (*)
(*) ・機能説明 + 手続き宣言部（指定なしの場合） (*)
(*) ・機能説明 + 手続き宣言部 + 型・定数・変数宣言部 (*)
(*) III) 使用説明書ファイル (*)
(*) IV) 検査仕様書・成績書ファイル (*)
(*) ② プログラムファイル (*)
(*) 整形化の程度 (*)
(*) ・同一ファイルに出力（指定なしの場合） (*)
(*) ・モジュール単位で別ファイルに出力 (*)
(*) ・手続き・関数・サブルーチン単位で別ファイルに出力

```

```

(*) ③ サブドキュグラムファイル (*)
(*) 整形化の程度 (*)
(*) ・同一ファイルに出力 (指定なしの場合) (*)
(*) ・モジュール単位で別ファイルに出力 (*)
(*) ・手続き・関数・サブルーチン単位で別ファイルに出力 (*)
(*)
(*) 5 ドキュグラム文法 (*)
(*) ① ドキュグラムは以下のように記述する。 (*)
(*) (* 区切り文字 仕様記述 *) (*)
(*) (* 仕様記述 *) (*)
(*) プログラム (*)
(*) プログラム (*)
(*) (* 仕様記述 *) (*)
(*) (*)
(*) (* 仕様記述 区切り文字*) (*)
(*) 但し、C言語の場合は '(*' と '*)' はそれぞれ '/' と '*' に (*)
(*) なる。 (*)
(*) ② 区切り文字を以下に示す。 (*)
(*) ① 外部仕様書 : 'B' (*)
(*) ② 設計仕様書 : 'D' (*)
(*) ・モジュール : 'M' (*)
(*) ・手続き : 'P' (*)
(*) ・関数 : 'F' (*)
(*) ・サブルーチン : 'S' (*)
(*) ・手続き (実行文) 開始フラグ : 'E' (*)
(*) (FORTRAN, C のとき開始側だけ記述する) (*)
(*) ③ 使用説明書 : 'U' (*)
(*) ④ 検査仕様書, 成績書 : 'C' (*)
(*)
(*) 6 ドキュグラム予約語 (*)
(*) 本システムは、1つ以上の空白または、EOL文字をデリミットとして (*)
(*) 1つの文字列を取り出し、それを予約語の候補としている。そのため、コ (*)
(*) ディング中に以下に挙げる文字列を記述する際は充分注意する必要があ (*)
(*) る。 (*)
(*) 以下に各言語の予約語をしめす。尚、予約語は全て半角の大文字でなけ (*)
(*) ればならない。 (*)
(*) ① PASCAL 言語の場合の予約語 (*)
(*) {, }, (*, *), {¥B, B¥}, {¥D, D¥}, {¥M, M¥} (*)
(*) {¥P, P¥}, {¥F, F¥}, {¥U, U¥}, {¥C, C¥} (*)
(*) (*¥B, B¥*), (*¥D, D¥*), (*¥M, M¥*), (*)
(*) (*¥P, P¥*), (*¥F, F¥*), (*¥U, U¥*), (*)
(*) (*¥C, C¥*), (*)
(*) MODULE, PROGRAM, PROCEDURE, FUNCTION, (*)
(*) DEFINE, DEFINE:, FORWARD, FORWARD:, EXTERN, EXTERN:, (*)
(*) LABEL, CONST, TYPE, VAR, (*)
(*) BEGIN, CASE, (*)
(*) END, END:, END. (*)
(*)
(*) ② C言語の場合の予約語 (*)
(*) {, }, /*, */ , /*¥B, B¥*/ , /*¥D, D¥*/ , (*)
(*) /*¥M, M¥*/ , /*¥F, F¥*/ , /*¥U, U¥*/ , (*)
(*) /*¥C, C¥*/ , /*¥E ¥*/ , (*)
(*) INT, SHORT, LONG, FLOAT, DOUBLE, CHAR, (*)
(*) #debug, #define, #eject, #else, #endif, #if, #ifdef, #ifndef, (*)
(*) #include, #line, #list, #module, #nolist, #section, #systype, (*)
(*) #undef (*)
(*)
(*) ③ FORTRAN言語の場合の予約語 (*)
(*) {, }, (*, *), (*¥B, B¥*), (*¥D, D¥*), (*)
(*) (*¥M, M¥*), (*¥F, F¥*), (*¥U, U¥*), (*)
(*) (*¥C, C¥*), (*¥E ¥*), (*)
(*) INTEGER, REAL, CHARACTER, *2, *4, *8, IMPLICIT, (*)
(*) PROGRAM, SUBROUTINE, FUNCTION (*)

```



第3図、「日本語による仕様書」作成支援システムの概要

4、支援システムの実現

DocuGramから、各種の仕様書 outputs する整形器を、第3図に示すように、高機能ワークステーション(Apollo Domain WS)とパソコン(NEC PC-9801)を使用したネットワーク(Ether)接続による分散システム上に実現した。

仕様の日本語入力とプログラム作成は、当初ワークステーション上の日本語ワープロをしようするつもりであった。この日本語ワープロは、ビットマップディスプレイやマウスを駆使する、先進的なソフトウェアであった。が、実際に使用してみると、極度に貧弱な操作性と性能のため、まったく実用に耐えないことが判明した。そこで、日本語入力についての方針を代え、パソコン上の日本語入力フロントエンドと汎用エディタを使用した。この日本語入力方式は、十分実用に耐えるものであった。作成したドキュラムは、ネットワークのファイル転送機能(FTP)を利用して

ワークステーションに送り、ワークステーション上でプログラムのコンパイル等を実施した。

整形器から出力された設計仕様書の部分を、第4図に示した。プログラムの静的定義部分(プログラム定義、定数・型定義等)が自動的に抽出されているのが明らかである。

5、おわりに 本システムによって得られる効果として、次のものが確認された。

- ・仕様書作成の効率化。
- ・プログラムと仕様の統合により、バージョンの不一致による相互の矛盾を排除する。
- ・仕様書の一部を、プログラムの静的定義部として有効利用することにより、仕様定義の重複的記述を排除す

ることが可能になり、プログラム作成の効率化が図られる。

- ・プログラムの静的定義部の早期確定は、プログラムの機能仕様の人手による確認にも役立つ。
- ・製型化機能により、仕様書の体裁を容易に整えることが可能。

末筆ながら研究の機会をあたえられた棟上昭男ソフトウェア部長・藤村是明情報システム研究室長、システム作成にあたられた富士通SSL 山口英紀氏に感謝する。

参考文献[1], 二木厚吉、実行可能仕様に基づく変換プログラミング、情報処理, V01.28 NO.7 July 1987

[2], D.E. Knuth, Literate Programming, The Computer Journal, Vol. 27 No.2 1984

[3], 大石東作、ソフトウェアの「日本語による仕様書」作成支援システム、情報処理学会第35回全国大会、3X-9,

第4図、整形器の設計仕様書（一部）

「日本語による仕様書」作成支援システム設計仕様書

1 概要

本仕様書は、電子技術総合研究所ソフトウェア部情報システム研究室において開発した「日本語による仕様書」作成支援プログラムの整型化部分について述べたものである。

開発言語はPascalを使用する。また、テンポラリファイルのリネームには、STREAM I/Oルーチンを使用する。

```
PROGRAM Compose(input, output, doc_file, bas_file, dsn_file,
                use_file, chk_file, pgm_file, sdc_file);
```

コンスタント定数

CONST

```
ResWdNo      = 116;          (* 予約語の数 *)
AlfaLen       = 11;          (* 文字列の最初の10文字を参照する *)
AlfaLen1      = 12;          (* 予約語文字列参照配列用配列要素数 *)
```

型宣言

TYPE

```
int16         = INTEGER;      (* 2バイトの整数タイプ *)
index         = INTEGER;      (* 配列添字 *)
int32         = INTEGER32;     (* 4バイトの整数タイプ *)
str80         = PACKED ARRAY [1..80] OF CHAR;
                (* 80バイトの文字列タイプ *)
str256        = PACKED ARRAY [1..256] OF CHAR;
                (* 256バイトの文字列タイプ *)
AlfaType      = PACKED ARRAY [1..AlfaLen] OF CHAR;
                (* 予約語の文字列タイプ *)
file_name_t   = str80;        (* ファイル名のタイプ *)
language_t    = (pascal, c_lan, fortran);
                (* 言語のタイプ *)
mp_f_t        = (modu, pfs, oth); (* 処理区分 モジュール単位／手続き単位 *)
fv_f_t        = (fun, cvt);    (* 詳細区分 機能のみ／全情報 *)
filekind      = (bas, use, chk); (* ファイル区分 *)
```

```
Symbols       =              (* 予約語シンボル *)
```

```
(LComSy, RComSy, LBraSy, RBraSy, LBasSy, RBasSy, LDesSy, RDesSy,
 LModSy, RModSy, LPreSy, RPreSy, LFunSy, RFunSy, LSubSy, RSubSy,
 LUseSy, RUseSy, LChkSy, RChkSy, EntSy,
 ProgramSy, ModuleSy, ProcedureSy, FunctionSy, ForwardSy, DefineSy,
 LabelSy, ConstSy, TypeSy, VarSy, BeginSy, CaseSy, EndSy, EndPSy,
 IntSy, RealSy, CharSy, LongSy, ShortSy, FloatSy, DoubleSy,
 SubroutineSy, DataSy, CommonSy, EquivalenceSy, ImplicitSy,
 IntegerSy, CharacterSy, LogicalSy,
 PreProcSy, OtherSy);
```

```

ResWd_t      = ARRAY[1..ResWdNo] OF str80;
              (* 予約語文字列配列型 *)
RwSy_t      = ARRAY[1..ResWdNo] OF Symbols;
              (* 予約語シンボル配列型 *)

```

```

(*****
(*)
(*) 変数宣言 (*)
(*)
(*****
VAR
(*-----*)
(*)
(*) 予約語配列 (*)
(*)
(*-----*)
ResWd      : ResWd_t;      (* 予約語文字列配列 *)
RwSy      : RwSy_t;      (* 予約語シンボル配列 *)
RwMk      : ARRAY[1..AlfaLen1] OF int16;
              (* 予約語文字列配列参照用配列 *)

(*-----*)
(*)
(*) 実行制御用各種フラグ (*)
(*)
(*-----*)
command    : str80;      (* 入力コマンド格納領域 *)
lan_f     : language_t;  (* 言語フラグ *)
spc_f     : BOOLEAN;    (* 仕様書出力フラグ *)
pgm_f     : BOOLEAN;    (* プログラム出力フラグ *)
sdc_f     : BOOLEAN;    (* サブドキュラム出力フラグ *)
bas_f     : BOOLEAN;    (* 外部仕様書出力フラグ *)
dsn_f     : BOOLEAN;    (* 設計仕様書出力フラグ *)
use_f     : BOOLEAN;    (* 使用説明書出力フラグ *)
chk_f     : BOOLEAN;    (* 検査仕様書・成績書出力フラグ *)
mp_f      : mp_f_t;     (* モジュール単位／手続き単位フラグ *)
fv_f      : fv_f_t;     (* 機能のみ／全情報フラグ *)
proc_name_sw : BOOLEAN;  (* proc_name必要時 TRUE *)
temp_out   : BOOLEAN;    (* テンポラリーファイル出力時 TRUE *)
pgntemp_sw : BOOLEAN;    (* テンポラリーファイルOPEN時 TRUE *)
sdctemp_sw : BOOLEAN;    (* テンポラリーファイルOPEN時 TRUE *)

```