5M-6

ＯＳ／ｏｍｉｃｒｏｎにおけるプログラム文書化支援環境の実現

大島利浩，並木美太郎，髙橋延匡
(東京農工大学，工学部，数理情報工学科)

１．はじめに

プログラム開発環境を整えるという点でプログラムの文書化は重要な意味を持っている．本稿では，ＯＳ／ｏにおける文書化支援環境について述べる．本研究室でも，日本語ワードプロセッサの利用によりプログラムとそれに対する文書の一体管理が進められている．しかし，バージョンアップやバグの修正などがあると，プログラムとの対応のとれた文書作成を行うにはかなりの時間が費やされてしまう．また，「能力のあるプログラマは文書を書かずにプログラムを作る」という事実があり，そのようなプログラマを支援し，労力を低減させるためには，効率の良い文書作成環境が必要である．こうしたことから，本システムは，計算機を利用して，作成されたプログラムからプログラム文書の生成を行うものである．

２．設計方針

プログラム文書化の基本はそのプログラムの持つ構造や関係を示すことである．具体的にはクロスリファレンスやコーリングストラクチャなどのツールが必要となる．以下にこうしたツールを含めたこの文書化環境の特徴を述べる．

(1)「ソースプログラムが最も信頼できる文書である」という考えから，言語Ｃのソースプログラムを利用して文書化を行う．

プログラムをデバッグ中にバグや修正の記録やバージョンアップなどによる変更記録などの文書を最も簡単に残すのなら，プログラム中にコメントにしてそれを記載すれば良い．また，コメントにして残さないにしても，変更などの記録は，プログラムを比較すれば情報を得ることができる．他のシステムでは，プログラム中に文書化用のコマンドを埋込み，それをプログラムから分離し，文書化を行っているものがある．本システムでは，こうしたことは行わない．

(2) コンパイラを文書化用データ生成に利用する．

コンパイラのパーザから出力される中間コードを主にそのデータとして利用する．当研究室で開発された日本語言語Ｃコンパイラ ＣＡＴは，コンパイラのような言語処理系としての位置付けだけでなく広くソフトウェアの開発支援環境を目的としている．具体的には，コンパイラとしての機能だけでなくデバッグの機能，文書化の機能をもＣＡＴは機能としている．すなわち，クロスリファレンス，コーリングストラクチャなど種々の文書を自動生成するツールを意図している．こうすることによる利点を次にあげる．

① ＣＡＴでは，プログラム中に日本語が使用可能である．そこで，それを利用することにより日本語の文書作成が可能となる．

② 構文解析の手間がいらないことと，構文解析の済んだことによるデータに対する情報が豊富であることがあげられる．具体的には，パーザから出力される中間コードには，関数や変数の詳細なデータ（関数や変数の参照情報，記憶クラス，データ型，サイズ，構造体のタグ名やメンバー名）が出力されている．

③ 仕様の固まった中間コードを使用することにより他言語のコンパイラはその中間コードを出力することだけで文書化環境がそのまま使用できる．つまり，この文書化環境は言語独立である．

３．文書化ツールとその機能

こうした環境の中のには以下のようなツールがある．

３．１　クロスリファレンス

プログラム中の識別子（関数名，変数名）の出現箇所を明示する．具体的には以下のような情報が得られる．

(1) データの型
(2) 定義されているファイル名
(3) リターン値のサイズ（関数の場合）
(4) 初期化の有無（変数の場合）
(5) 関数のコーリングシーケンス（関数の場合）
(6) 参照している関数名とそのファイル名
(7) 参照されてい箇所でのリターン値や引数の状況（リターン値や引数のサイズ）

３．２　コーリングストラクチャ

言語Ｃの関数の階層を生成するものである．ある関数はどこで定義されいるか，その関数内のどこでどんな関数を呼んでいるかを示す．

３．３　ソースプログラムからの情報

中間コードからだけではどうしても得られない情報がある．たとえば，行番号やマクロなどの情報は中間コードの現在の仕様では得ることができないので，プリプロセッサの段階で情報を引き出し，クロスリファレンスで利用することを考えた．また，関数の機能，変数の説明など意味情報は，プログラム中にコメントにして記載されるとプログラムの理解には非常に役立つ．そのため，こうした文書はプログラム中にコメントとして記載されていることが多い．そこで，このコメントを，プログラム本体からの情報として取り出し，内部仕様書作成に利用する．そうする場合にも，自由な形式で書かれているコメントをそのまま利用するのは非常に困難なので，コメントの自由度を失わない程度の，雛形を設けて利用

An Implementation of Computer Assisted Software Documentation on OS/omicron

Toshihiro OHSHIMA, Mitarou NAMIKI and Nobumasa TAKAHASHI
Tokyo University of Agriculture and Technology

```
/*************************************************/
/*<システム名>      : crsrf :                    */
/*<ファイル名>      : main.c :                   */
/*<作成者>          : 大島 :                     */
/*<作成日>          :1987/12/31:                 */
/*<モジュール仕様>  :  モジュールの説明の記述  : */
/*-----------------------------------------------*/
/*<Rev>             :1.10:                       */
/*<改訂者>          :大島:                       */
/*<改訂年月日>      :1988/3/05:                  */
/*<改訂内容>        :    改訂内容の記述    :     */
/*-----------------------------------------------*/
```

モジュールごとのコメントの記述形式

```
/*-----------------------------------------------*/
/*<関数名> : function :                          */
/*<引数   > CHAR    *s;:    引数の意味の記述  : */
/*<引数   > INT     n; :    引数の意味の記述  : */
/*<関数値> :        関数値の意味の記述        : */
/*<機能   > :       関数の処理内容の記述      : */
/*<注意   > :       注意事項の記述            : */
/*-----------------------------------------------*/
```

関数の機能説明のコメントの記述形式

図１　コメントの雛形

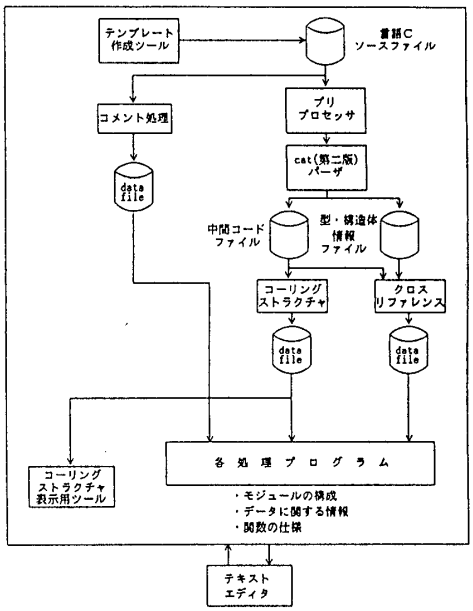

図２　システム構成図

しやすくすることを考えた（図１参照）．また，この雛形に沿ったコメントをプログラム中に効率良く書いてもらうためのテンプレート作成ツールがある．

４．システム構成

システム構成を図２に示す．現在ＯＳ／ｏ上にはクロスリファレンスと，コーリングストラクチャ，それとコメントやマクロなどの情報を引き出すコメント処理ツールがある．前の二つは，２パス構成になっており，パス１では，パーザから出力される中間コードを入力データとし，コンパイル中にそれらを実行することによりモジュールごとのデータを作成する．このあと作成されたデータをまとめてパス２に通すことにより，システム全体のデータが作成される．こうして作成されたデータは，ツールを通すことにより，ＬＢＰを使用して紙面上の文書として出力される．クロスリファレンスでは，コメントやマクロの情報を付加し，これをまとめてプログラムの仕様書が作成できる．図３は，コーリングストラクチャの出力結果である．

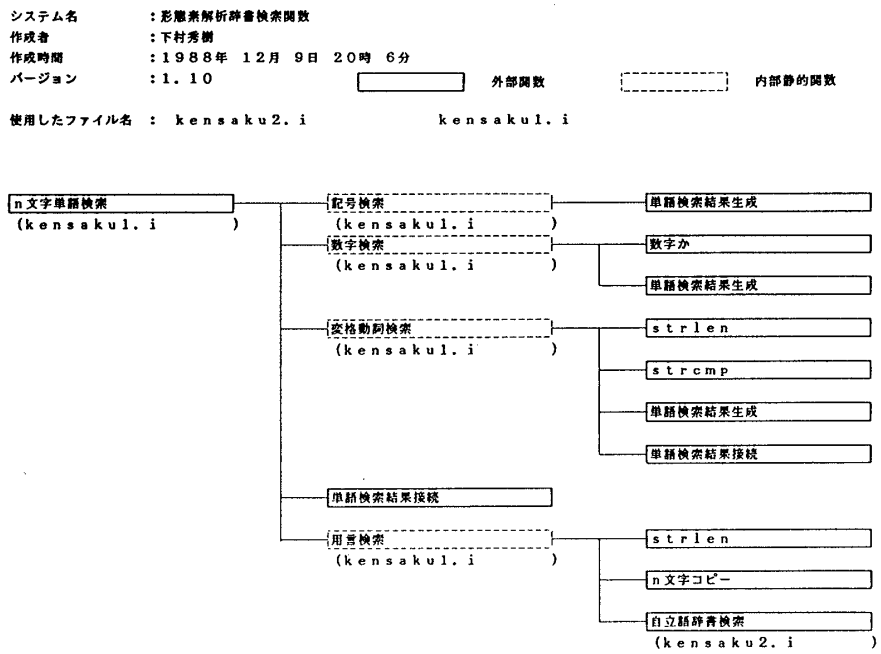

図３　コーリングストラクチャ

５．課題

現時点では紙面上に文書が作成されるが，作成される文書量は非常に多く，実際のデバッグやバージョンアップにおいては，必要な情報だけを選択して利用できる文書データベースのような，対話型のインタフェースを持った環境が必要であると考えられる．

６．おわりに

本稿では，日本語プログラミング環境であるＯＳ／ｏ上での言語Ｃコンパイラ ＣＡＴを利用した文書化環境について述べ，(1) 計算機での文書化の有効性，(2) コンパイラを利用した文書化の有効性，(3) 計算機での文書化において日本語が使用できる環境の重要性などの成果が得られた．

参考文献

[1]並木美太郎，他，“言語Ｃコンパイラcatの方式設計” ソフトウェア工学研究会 ４８－２，１９８６年６月