

MOSDENV

2B-2

(保守を考慮したソフトウェア開発環境)

におけるドキュメンテーション方式

長田 和久 田中 英彦 元岡 達

(東京大学工学部)

1. はじめに

増大するソフトウェアコストを低減するために、MOSDENV (Maintenance Oriented Software Development Environment) [仮称] を計画している。これは、図1のような改版が数多くなされるような場合を想定している。すなわち、初期投資が多少割高でもトータルでひきあえばよいことをねらう。(図1では、従来、“保守”に一括して分類されていた部分を version up としている。)

2. MOSDENVの概要

予防保守をしつつソフトウェア開発を支援する環境—MOSDENV—は、要求分析/仕様化技術、ドキュメンテーション方式、ソフトウェア開発方法論などをもとに保守用データベース、各種支援ツールなどを有機的に組み合わせて構成される。

トータルのコストを下げるために、「要求分析/仕様化」を強力に支援し、保守(改版)を容易にする。その中心をなすドキュメンテーション方式は、機能分割・段階的詳細化に適合するような階層構造をもつものとする(図3)。システム構成は(図2)、VAX/UNIXの環境のもとに保守用データベースとワークステーションとの結合形態をとり、マルチウィンドウ機能を活かしたエディタ Emacs (図4)をはじめ、各種ツール群を用意する。

保守支援されるソフトウェア記述言語は、“C”、“lisp”および“C-shell script”である。また、MOSDENV自身もMOSDENVで記述・保守 self-descriptive, self-maintainable をめざす。

*長所

- ・ライフサイクルをよく表現する
- ・設計思想、製品を軸とする(軸の太さで表す)
- ・関連製品の開発、設計方針の変更などを表現しやすい
- ・各段階のコストを表現しやすい(軸からの距離)
- ・経過時間を表現しやすい(軸に平行な方向)
- ・各版の各段階の相互関係を表現しやすい
- ・改版が繰返される場合に適する

*欠点

- ・図が描きにくい
- ・3次元的感觉を要する [螺旋型モデル]
- ・並行開発を表しにくい

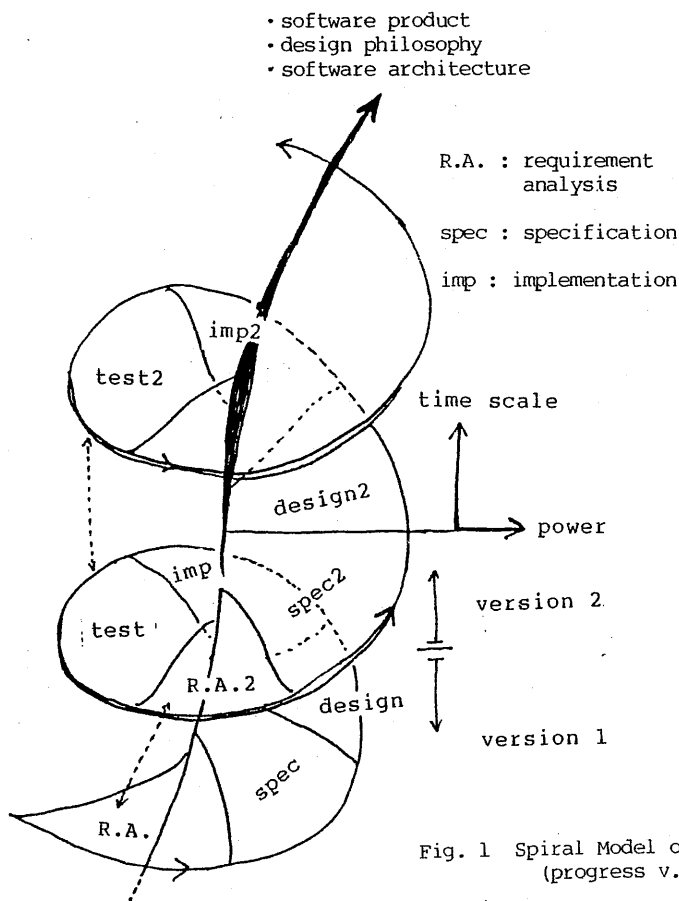


Fig. 1 Spiral Model of Software-Life-Cycle (progress v.s. manpower)

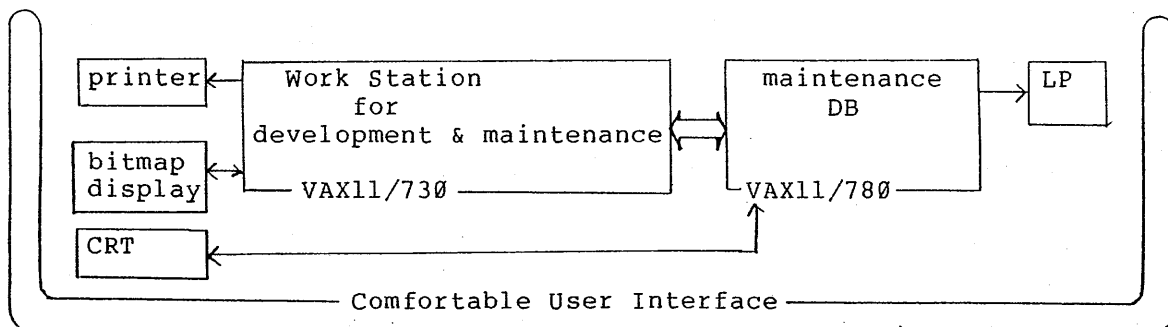


図2 システム構成

3. ドキュメンテーション方式

図3に示すように、トップダウン設計によって作られた各モジュールごとに、仕様書、ソースコード、マニュアル、テストデータなどを用意するが、これらを統一的に取り扱うことにより、ドキュメント群の管理を容易にしている。すなわち、モジュール、バージョン、種別(表1)によって細分したドキュメントファイルを作る。“Table”, “Where”を参照することにより、

- ・プロダクト全体の解説書やマニュアル
- ・あるレベルまでの仕様書
- ・あるモジュールに関するすべてのドキュメント
- ・通常のソースリスト
- ・プロダクトの設計メモ

を取り出したり新たに作成することが可能である。さらに、これらの操作を過去の版について行なうこともできる。

```

overview : rough sketch of the module
description : functional description of the module
spec      : specification of the module
memo      : design memo of the module
source    : source program of the module (source.c)
pcl       : program check list (test data)
interface : condition of the module as a blackbox
manual    : manual of the module, usage description (manual.roff)
object    : compiled object (source.o)
Table     : elements of the lower modules
Where     : elements of the upper modules
History   : check of the time-dependency
    
```

Table 1. Attributes of a document
(in Document Formula)

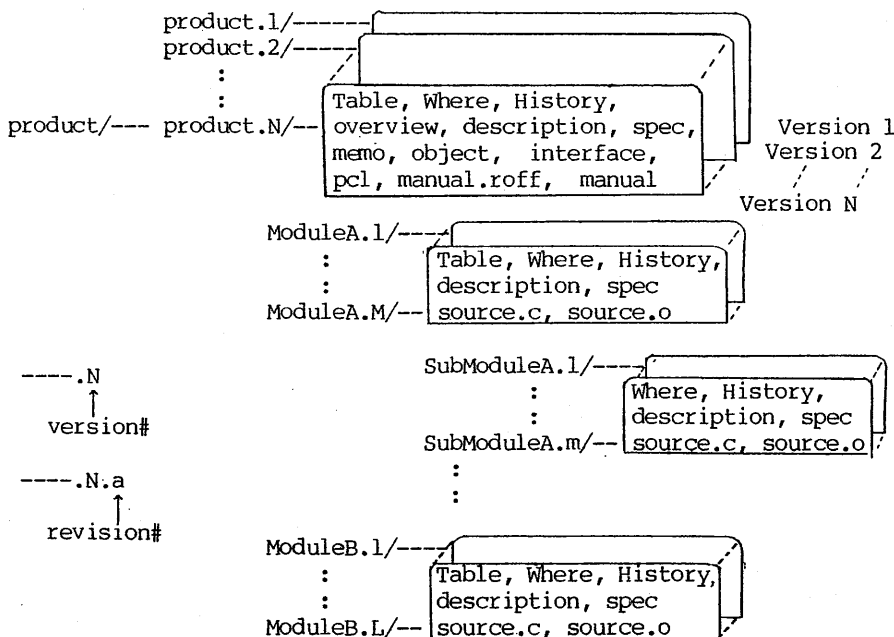


Fig. 3 An Example of the Document Formula

Note 1: the content of Table
e.g. product.2/Table → ModuleA.2
ModuleB.1.b

MOSDENVにおいてドキュメント作成時には、次の3つのモードがある。(図5)

- I: init. 初期開発モード。システムの適切な誘導により次々にドキュメントを編集する。top-down design, step-wise refinement を支援するようなprompting systemを用意 (lispによる)。
- D: debug, test & debugモード。バグ取りのために、time-dependency によらず、ソース等を変更してテストできる。(rapid-prototypingを支援)。
- R: revise. 改版モード。改版のために、ドキュメントを変更する際、module-dependency, time-dependency に応じた順序に従ってドキュメントを変更できる。また、変更理由“Why”等の記述も支援する。

IモードとDモードの終了時には、time, module-dependency などの各種情報がチェックされる。これによりドキュメント間の consistency を保つ。

現在、ドキュメント群を操作する各種ツール群 (construct, revive, purge, crossrefer, etc) を作成している。

4. おわりに

MOSDENVは、研究途上であるが試作品、中間評価ができれば順次発表してゆきたい。

ソフトウェア開発の実用的な手法について御教授いただいた、日立製作所の野口健一郎氏に感謝する。

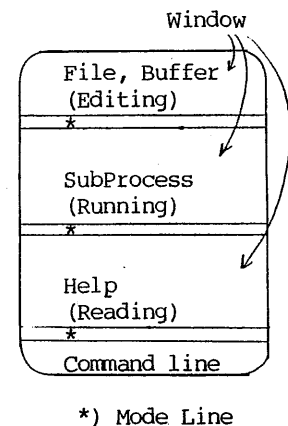


Fig. 4 windows of Emacs (Example)

	I	D	R
description	1	*	1
spec	2	*	2
source	3	*	3
object	4	*	4

Fig. 5 Mode and time-dependency