ソフトウェア文書のためのエディタ—Spec†

堀川博史†† 伊藤俊之†† 高野彰††

ソフトウェアの生産物はプログラムだけでなく各種文書も含まれる．プログラミングには多くの時間が費やされるが，文書作成にはそれ以上の時間が費やされる．設計段階では，文書作成に要する時間は 50% 以上になっている．そこで，筆者らは容易にソフトウェア文書を作成するためのツール，仕様書エディタ Spec (Specifications Editor) を提案する．当エディタの設計に当たって，筆者らは通常のオフィス文書との比較からソフトウェア文書の特徴を洗い出した．最も大きな特徴としては，ソフトウェア文書はトップダウンアプローチで作成されることにある．この観察に基づいて，ソフトウェア文書の枠組みをあらかじめ定義として与えられると，その定義に従って文書編集を支援する方式をとっている．文書の枠組みは文書アーキテクチャにおける汎用論理構造（章，節，項の構造）である．そして，ソフトウェア文書のための幾つかの属性を汎用論理構造の要素である論理構成要素に付加した．これらは，定形文章の自動記述，記述内容に関するガイダンス，自動複写，内容の検査である．また，仕様書エディタの適用を通しての評価について報告する．仕様書エディタは文書作成作業における 30% の記述を自動的に記述することができた．さらに，記述量の削減のために，仕様書の流用を目的として，ソフトウェア文書を論理構造単位で管理することにより，論理構造単位で他文書の引用ができる機能を実現した．

1. はじめに

今日における計算機の使用目的の一つに，メモ，手紙，報告書などの文書の作成があげられる．ソフトウェア生産においては，プログラムコードだけでなく，外部仕様書，内部仕様書，検査仕様書など多くの仕様書が作成されている．プログラムコードの生産物の全体に占める割合は，ある調査によると 5% から 10% の間にあった[1]．生産物の大部分は文書の形態をとり，ソフトウェア製造に要する費用の半分近くは仕様書の作成に費やしている．近年，ワードプロセッサの進歩と普及により，文書作成の効率化が図られてきた．しかし，仕様書の記述に対しては多くの規則があり，これらの規則を支援するエディタが望まれる．そこで仕様書エディタ Spec (Specifications Editor)[2),3)] を UNIX ワークステーション上に試作した．仕様書エディタはソフトウェア文書の枠組みをあらかじめ構造定義として与えておいて，その定義に基づいて文書編集を支援する．構造を基に作動するエディタは，プログラミング環境のために，Mentor[4)]，Cornell Program Synthesizer[5)]，Gandalf[6)]，Poesy[7)] などが報告されており，文書処理のために Tioga[8)]，Grif[9)] などが発表されている．文書の構造は大きくレイアウト構造と論理構造という二つの局面をもつ[10)]．レイアウト構造は，文書をページの集まりとして扱うのに対して，論理構造は，文書を章，節，項の集まりとして処理する．文書の枠組みは文書アーキテクチャにおける汎用論理構造（章，節，項の構造）である．Tioga, Grif などこの観点に基づく汎用的な木構造エディタと見なすことができる．一方，仕様書エディタではソフトウェア文書の記述規則のための幾つかの属性は汎用論理構造の要素である論理構成要素に付加した．これらは，定形文章の自動記述，記述内容に関するガイダンス，自動複写，内容の検査などである．

以下，2章ではソフトウェア文書の特徴について，3章では仕様書エディタの機能について，4章では仕様書エディタの実現について，5章では評価と改良について述べる．

2. ソフトウェア文書の特徴

ソフトウェアの開発工程を計画段階，設計段階，製造段階，試験段階，運用・保守段階の5段階に分けて捉える．設計段階をさらに，システム仕様設計，外部仕様設計，内部仕様設計の3段階に区分する．それぞれの設計段階ではシステム仕様書，外部仕様書，内部仕様書という文書が作成される．システム設計では，実現するシステムの仕様を明確にし，機能ブロックに分割し，ハードウェア，ソフトウェアの分担を明確にし，機能ブロック構造を決定する．外部設計では，各機能ブロックの仕様を定義し，それぞれのプログラム構造を決定する．内部設計ではプログラムの仕様を定

† An Editor for Software Documents—Spec by HIROSHI HORIKAWA, TOSHIYUKI ITOH and AKIRA TAKANO (Systems & Software Development Department, Information Systems & Electronics Development Laboratory, Mitsubishi Electric Corporation).

†† 三菱電機(株)情報電子研究所システム・ソフトウェア開発部

義し，それぞれのモジュール構造を決め，各モジュールの仕様を定義する．各仕様書には記述対象の目的，位置付け，機能，入出力情報，使用法，操作員インタフェース，エラー処理等の仕様を記述する．

ソフトウェア仕様書は通常のオフィス文書と比較して次の違いをもつ．

① トップダウンアプローチで作成される．

② 記述量が多い．

③ 記述規則が多い．

④ 他の仕様書（特に前段階で作成される仕様書）との間の関係付けが行われる．

⑤ 一冊の仕様書の中での記述順序は記載順序と一致しない．特に，懸案事項を残して先に進むことが多い．その結果，修正が多く発生する．

⑥ 文章の代わりに，図・表で表現することが多い．

⑦ 作成されるソフトウェアの適用分野の違いなどにより記述規則の統一が難しい．

3. 仕様書エディタの機能

ソフトウェア文書と通常のオフィス文書の上述した違いを考えると，通常のワードプロセッサだけでは十分な支援を得ることができない．そこで，ソフトウェア仕様書作成のために次の機能を実現することにした．

（a） ソフトウェア文書の枠組みにそって文書を作成する機能．

（b） 仕様書の標準化と記述量軽減のために定形的な部分を自動記述する機能．

（c） 個人差の解消と作成時間の短縮のための記述すべき内容に対するガイダンス機能．

（d） 他の仕様書との関係付けと記述量の軽減のための自動複写機能．

（e） 同一記述の記述ミス，修正誤りを防ぐための矛盾検出機能．

（f） 記述もれを防ぐための未完成部管理機能．

（g） 図・表の編集機能．

これらの機能の具体的仕様をエディタの中に作り付けにすると，1種類の仕様書に対してだけしかエディタを適用できない．また，同じ種類の仕様書でもそのソフトウェアの適用領域の違いにより異なる記述規則をもつことがある．そこで機能（b)～（e）は，記述規則として外部から与える形で実現した．その結果，仕様書エディタが扱える仕様書の形式は自由度をもつ．

（1） 自動記述機能

仕様書の記述項目（章立て，節立て，項立て）は通常決まっている．そこで，章節構造を定義として仕様書エディタに与えると，仕様書エディタは章節項の見出し行を自動記述する．その結果，仕様書エディタを用いて作成される仕様書はこの章節構造をとることが保証される．章節項の見出し行以外にも典型的な文章を自動記述するように指定しておくことができる．例えば，エラー処理の章では『システムは検出できるエラーをすべて処理し，エラーメッセージを表示する．…』という文章から始まる．これらの定形文章は自動記述される．

（2） ガイダンス機能

仕様書に何を記述するかは，仕様書エディタがガイドする．例えば，プログラム名を入力すべきところでは，『プログラム名を入力して下さい』というメッセージが出る．ガイドメッセージは定義として与えておくことができる．

（3） 自動複写機能

仕様書の記述の中には同じ内容をもつ部分がある．例えば，『機能一覧』にも『機能単位の処理概要』にも同一の機能名が記述されなければならない．同じ記述を繰り返すことは利用者の負担になるので，仕様書エディタはそれらを自動複写する（**図1**参照）．

システム仕様書の中には，そのシステムで使用される共通ソフトウェア群の記述があり，それらは下位の仕様書への自動複写の対象となる．しかし，システム中のあるプログラムがすべての共通ソフトウェアを使用するとは限らない．そのようなプログラムに対する仕様書の作成状況を想定し，共通ソフトウェア等の自動複写時に，個々の対象を複写するか否かを利用者が指定することもできる．

（4） 矛盾検出機能

自動複写が行われた場合は，複写先と複写元の間で

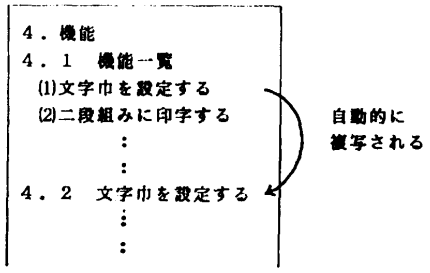


図1 自動複写機能
Fig. 1 An automatic copy function.

記述内容が同一でなければならないという表明が自動的に設定される．これにより，例えば，『機能一覧』と『機能単位の処理概要』の修正忘れや修正ミスを防ぐことができる．

また，記述が制限に従っているかどうかの検査が行われる．記述の制限には，文字列長，頭文字，文字列に対して行える．

（5） 未完成部管理機能

仕様書エディタは懸案事項であることを示す記号を検出し，その文章を含む論理構成要素を未完成部として報告する．また，文書内容をもたない論理構成要素も検出する．

（6）その他の機能

文章処理の基本機能である文章の挿入，削除，貼り付け，置き換えに加えて，線，円，長方形等の基本図形編集機能のほか，データフロー図の編集が可能である．

4. 仕様書エディタの実現

（1） 文書アーキテクチャ

文書は文書内容とそれらを特徴付ける論理構造とレイアウト構造から構成される[10]．文書内容は文字列，図形等の記述される内容である．論理構造は章，節，項の構造を管理する．レイアウト構造は文書内容が文書中にどのようにレイアウトされるかを管理する．レイアウト構造の管理単位はページ，ページ内位置情報等である．同種の仕様書は同形の論理構造をもつ．そこで，文書は文書クラスの一派生例として捉える．文

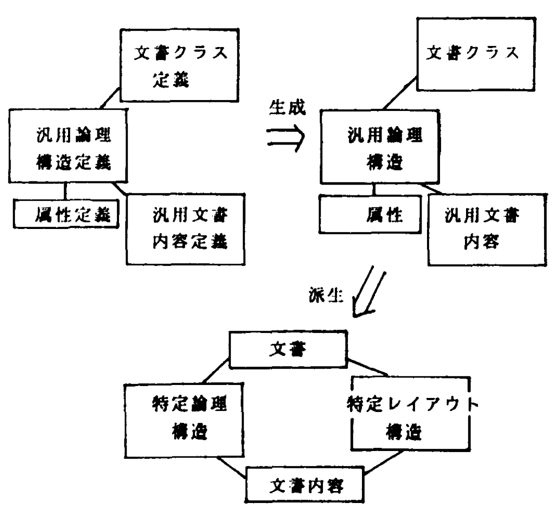


図 2 文書アーキテクチャモデル
Fig. 2 The document architecture model.

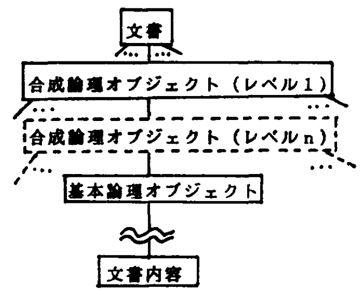


図 3 論理構造と文書内容
Fig. 3 Logical structure and document contents.

書クラスの論理構造を汎用論理構造，文書の論理構造を特定論理構造とよび区別する．文書クラスは文書クラス定義により定義される（図2参照）．文書クラス定義は，汎用論理構造定義，汎用文書内容定義，属性定義をもつ．

論理構造は木構造をしている．木のノードをオブジェクトとよぶ．木構造の枝はオブジェクトの一つ下のオブジェクトへの分割を示す．オブジェクトには，下にオブジェクトをもつものと，文章内容をもつものがある．前者を合成論理オブジェクトとよび，後者を基本論理オブジェクトとよぶ（図3参照）．

（2） 文書クラス定義

仕様書の記述規則を定義するためには文書クラス定義を用いる．文書クラス定義は属性文法（属性付きBNF）で定義する．図4に文書クラス定義の構文を示す．仕様書エディタは文書クラス定義を解析し，文書クラスを生成する．汎用論理構造定義と汎用文書内容定義は文書クラス定義の BNF 部で指定され，属性定義は属性部で指定される．汎用論理構造は汎用論理構造定義における基本論理オブジェクト名を葉とし，合成論理オブジェクト名を節とした木構造である．図5に文書クラス定義の例を，図6に対応する汎用論理構造を示す．

```
文書クラス定義：：＝｛生成規則 ｝＊
生成規則      ：：＝ ＢＮＦ部 ｛ 属性部 ｝
ＢＮＦ部      ：：＝ 合成論理オブジェクト名 ＝
                  ｛ 基本論理オブジェクトの種別
                     基本論理オブジェクト名
                     ［ ［ 汎用文書内容 ］ ］ ｝＊
                  ｛ 番号のパターン［ ！ ］
                     合成論理オブジェクト名 ｝＊
属性部        ：：＝｛ 属性規則 ｝＊
属性規則      ：：＝ 番号 ＝ ｛ 属性関数 ｝＊

｛ ｝＊は繰返し ［ ］は省略可能
```

図 4 文書クラス定義の構文
Fig. 4 Grammar of documentation class definition.

```
外部仕様 = ００概要 ００構成 ００設計条件 ００機能
           ００入出力情報 ００エラー処理
           ００外部インタフェース ００実現方式
           ００その他 ００関連資料
                                    ( )
                :
                :
機能 = １機能［機能］ ００機能一覧
       ００！機能単位の処理概要          ( )

機能一覧 = １機能一覧［機能一覧］ ０１！機能項目 ( )

機能項目 = １機能項目
          (1＝$ＰＲ(.,［機能名］))

機能単位の処理概要 = １処理項目
    (1＝$ＣＰ(機能項目.１,ＯＭ)＋
        $ＰＲ(.,［正常パターンにおける概略機能］))
                :
                :
```

図 5 文書クラス定義の例
Fig. 5 An example of documentation class definition.

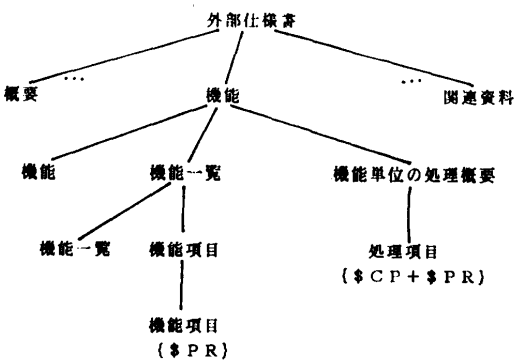


図 6 汎用論理構造 (図5に対応)
Fig. 6 A corresponding generic logical structure of Fig. 5.

BNF 部の右辺に1桁の数字を付けて現れる名称は基本論理オブジェクト名である．1桁の数字は基本論理オブジェクトの種別を示す．『1』は文章，『2』は図形，『3』は表，『4』はデータフロー図を示す．図5では機能，機能一覧等の文章オブジェクトが定義されている．

仕様書エディタは仕様書中の文書内容と基本論理オブジェクトの関係を管理する．これらの文書内容には①汎用文書内容により自動記述されるもの，②章，節，項に対して自動的に生成される番号，③自動複写属性により他の位置にある文書内容が自動転記されるもの，④利用者が入力した文書内容，がある．①と②は文書クラス定義の BNF 部で指定され，③は属性部で指定される．

汎用基本論理オブジェクトが汎用文書内容をもつとき，仕様書エディタはこの文書内容を関連する特定基本論理オブジェクトが生成されたときに生成する．図5の基本論理オブジェクト『機能』における『[機能]』は汎用文書内容定義である．

BNF 部の右辺に2桁の数字を付けて現れる名称は合成論理オブジェクト名である．2桁の数字は節，項の番号付けのパターンを指定する．図5において『機能単位の処理概要』は『00』なので『1.』『1.1.』『1.1.1.』…，『機能項目』は『01』なので『(1)』『(2)』『(3)』… である．

属性関数 $CP は関連した文書内容を自動的に複写する．この関数は複写元が繰返し可能な場合オプションをもつ．Oオプションは順次複写し，Sオプションは選択的に複写する．属性関数 $EC は前段階で作成された文書から自動複写する．図5の基本論理オブジェクト『処理項目』に対する $CP は『処理項目』が生成されたときに基本オブジェクト『機能項目』の文書内容が複写される．

ガイダンスを行う属性関数 $PR は引数として与えられた文字列をガイドメッセージとする．この属性関数の引数は文字列だけでなく，文字列と同時にある基本論理オブジェクトの指定を書くことができる．この場合，指定された基本論理オブジェクトに関連付けられている文書内容と与えられた文字列とでガイドメッセージを合成する．

文字『！』をもつ合成論理オブジェクトは，繰返し可能を示す．繰返し可能は1個の汎用合成論理オブジェクトに対して1冊の文書の中で複数の特定合成論理オブジェクトが派生可能である．図5において，『機能項目』に対応する利用者入力がある限り，その特定基本論理オブジェクトが生成される．また，『処理項目』は『機能項目』から自動複写を行う属性をもち，複写元の『機能項目』に対応する特定基本論理オブジェクトがあるだけ『処理項目』に対応する特定基本論理オブジェクトが生成される．

（3） 仕様書エディタの構造

図7に仕様書エディタのプログラム構造を示す．仕様書エディタは，文書クラス定義処理プログラム，編集プログラム，検査プログラムの3本のプログラムからなる．

文書クラス定義処理プログラムは文書クラス定義から汎用論理構造，汎用文書内容，属性をもつ文書クラス（仕様書のスケルトン）を生成する．

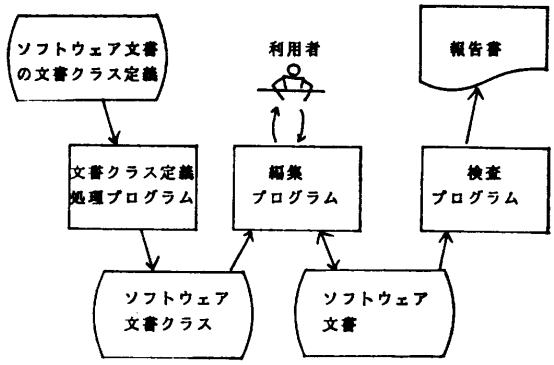


図 7 仕様書エディタの構造
Fig. 7 Structure of Spec.

編集プログラムは文書クラスから特定論理構造を派生させながら，日本語文書を処理する．また，属性関数に基づきガイドメッセージの表示，自動記述，自動複写等を行う．

検査プログラムは記述内容に付けた制限検査，および同一内容である部分の検査を行うと共に，仕様書のでき上がり具合（分母は論理構造構成要素の数）を表示する．

（4） 画面の例

図 8 は編集プログラムのレイアウト処理ウィンドゥの例である．最初の2行，章『4』と節『4.1』の見出し行は汎用文書内容により自動記述されたものであり，次の2行，『(1),(2)』に続く記述は利用者が記述したものである．『4.2』と『4.3』の見出し行は属性により自動複写されたものである．『4.3』の節題には修正が施されている．画面中央の小さなウィンドゥはガイドメッセージウィンドゥである．

図 9 は編集プログラムの論理構造処理ウィンドゥである．このウィンドゥは目次の働きをもつ．画面左は文書の木構造を示し，画面右は実際の章・節・項題の文字列である．

図 10 は検査プログラムの出力例である．未実行の属性，同一性違反の抽出，未派生のオブジェクトの表示の後，でき上がり具合を表示している．

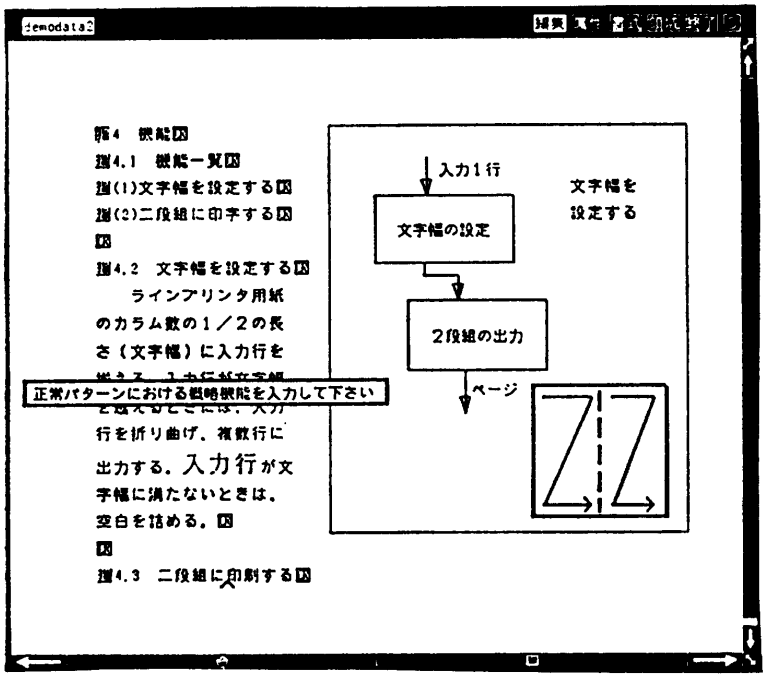


図 8 編集プログラムのレイアウト処理ウィンドゥの例
Fig. 8 An example of layout processing window of the editor.

5. 評価と改良

（1） 実施形態

実際に仕様書を作成することで評価を行う．対象としたプログラムの規模は約 1.5k ステップであり，文字列操作，C言語用ツールなどの3個のシステムプログラムの外部仕様書3冊である．

（2） 定量的な効果

次に，幾つかの定量的なデータを示す．

- 外部仕様書の定義の規則数：70 規則
- 外部仕様書の行数（3冊の平均）
  - 全体（空行を含む）： 313 行
  - 利用者の入力した行： 222 行（71%）
  - 自動記述された行 ： 70 行（25%）
  - 自動複写された行 ： 12 行（ 4%）
- 外部仕様書の文字数（3冊の平均）
  - 全体（タブ，改行，図形要素は1文字と数える）： 3412 文字
  - 利用者入力文字 ： 2365 文字（69%）
  - 自動記述された文字： 892 文字（26%）
  - 自動複写された文字： 155 文字（ 5%）

ここに示した数字からは，約 1.5 k ステップのプログラムに対する外部仕様書作成という工程では，約 30

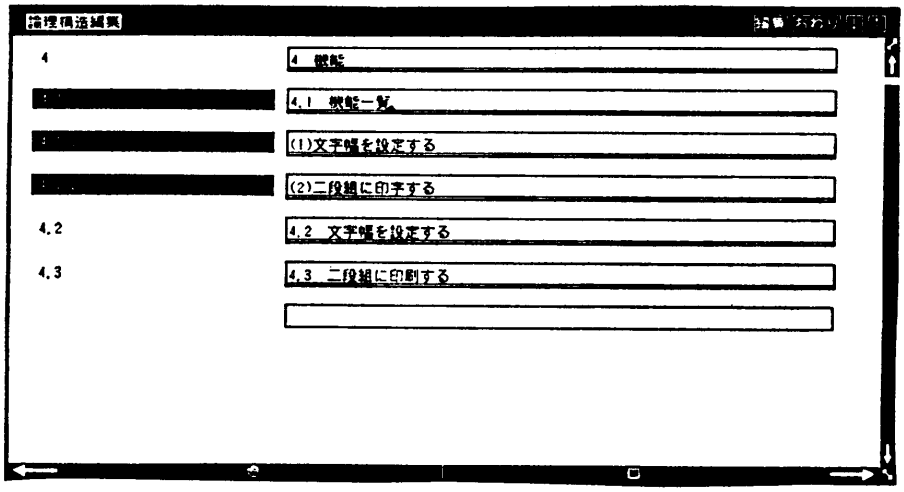


図 9 編集プログラムの論理構造処理ウィンドゥの例
Fig. 9 An example of logical structure processing window of the editor.

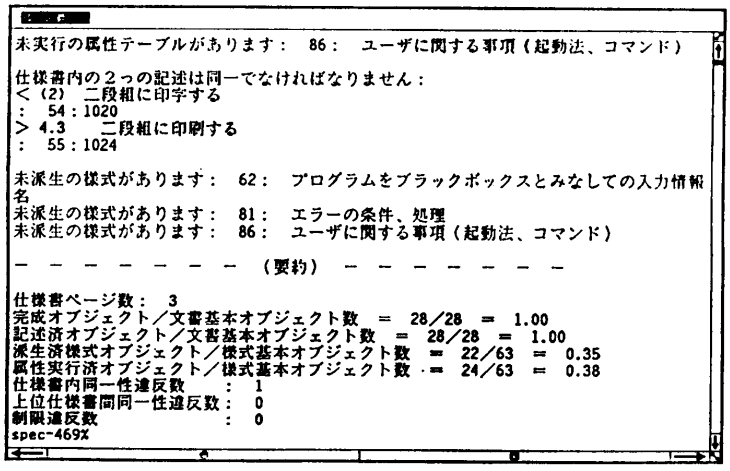


図 10 検査プログラムの報告書の例
Fig. 10 An example of report of document checker.

%の入力作業の軽減が行われるといえる．

（3） 属性関数の評価

●ガイダンス関数

一般にエディタのガイダンスといえば，『ドライブAにフロッピーを入れてください』といった操作に対するガイダンスである．これに比べて仕様書エディタでは記述すべき内容のガイダンスを行う．この効果は定量的には出てこないが，便利である．

当初，予想しなかった使い方としては，一つ前の記述に対するレビューチェック項目を表示させると，利用者がセルフチェックしながら仕様書を記述できる．

●自動複写機能

1冊の仕様書の中でも，全体説明があって，その後に各項目の詳細説明が行われるパターンが多い．このパターンは，外部仕様書の中では，ハードウェア構成，ソフトウェア構成，機能，モジュール構成，データなどで用いられる．しかし，実際にはこれらの一部のものにしか自動複写は適用できなかった．これは全体説明の記述法として図で記述されることが多いことによる．

（4） 改良

記述量の抜本的削減を図るためには，仕様書の部品化が一番である．そこで，仕様書の例文を参照して作成できるようにした[11]．

例文文書の管理に関しては文書クラスごとに論理構造を文書の枠組みとして用意し，その中に複数の例文を細文化してもつ形式で実現した．例文は，章，節，項等の論理構造要素単位で複写することができる．

例文の膨張を加味し，参照単位はファイルで保存し，論理構造の枠組みはオペレーティングシステムのディレクトリを利用している．

6. おわりに

ソフトウェア生産における生産物の大部分は文書の形態をとる．そこで，ソフトウェア仕様書の作成支援を目的として，仕様書エディタを提案した．仕様書の記述に対しては多くの規則があり，仕様書エディタではこれらの規則を自然に遵守させるようにする．特に，仕様書はトップダウンアプローチで作成されることから，ソフトウェア文書の枠組みをあらかじめ定義

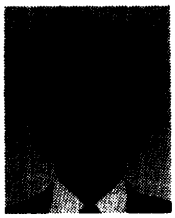
として与えておくことにより，仕様書エディタは，その定義に基づいて文書編集を支援する．文章の枠組みは文書アーキテクチャにおける汎用論理構造である．そして，この汎用論理構造の要素である論理構成要素に，ソフトウェア文書の記述規則のための幾つかの属性を付加した．これらは，定型文章の自動記述，記述内容に関するガイダンス，自動複写，内容の検査などである．その結果，仕様書エディタの自動記述機能，自動複写機能により文書作成作業における 30% の記述を自動的に記述することができた．このことから，仕様書エディタを用いることによってソフトウェア文書作成作業の標準化・効率化が期待できる．今後，ソフトウェアの仕様書に限らず，特許明細書等の他の定形文書に対しても適用していきたい．

謝辞 仕様書エディタの試作に際しては，三菱電機(株)情報電子研究所の古川栄司氏，藤本卓也氏，行徳孝彦氏の協力を得た．仕様書エディタの母体となる文書処理プログラムは同所の坂下善彦氏，土田泰治氏，田中朗氏によるものである．ここに記して深謝申し上げます．

参 考 文 献

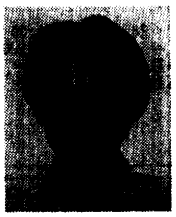
1) Furuta, R.: Trends in Interactive Preparation of Documents, *Compcon '87*, pp. 251-254 (1987).
2) 高野，大川，春原：仕様書作成支援ツール，第27回情報処理学会全国大会論文集，pp. 473-474 (1983).
3) 堀川，古川，高野，春原：仕様書エディタ：SPEC，情報処理学会ソフトウェア工学研究会，43-3, pp. 1-8 (1985).
4) Donzeau-Gouge, V., Kahn, G., Lang, B., Melese, B. and Morcos, E.: Outline of a Tool for Document Manipulation, *IFIP '83*, pp. 615-620 (1983).
5) Teitelbaum, T. and Reps, T.: The Cornell Program Synthesizer: A Syntax Directed Programming Environment, *Comm. ACM*, Vol. 24, No. 9, pp. 563-573 (1981).
6) Krueger, C. W.: The GANDALF System Reference Manuals, Technical Report, p. 100, Computer Science Department, Carnegie-Mellon University (1986).
7) 宮本，浅見：プログラム構造に基づいた編集機能をもつテキスト・エディタ，情報処理学会論文誌，Vol. 20, No. 6, pp. 474-480 (1979).
8) Teitelman, W.: A Tour through Cedar, *IEEE Softw.*, Vol. 1, No. 2, pp. 44-73 (1984).
9) Quint, V. and Vatton, I.: GRIF: An Interactive System for Structured Document Manipulation, in *Text Processing and Document Manipulation* (van Vliet, J. C. ed.), pp. 200-213, Cambridge University Press (1986).
10) ISO/IS 8613: Office Document Architecture (ODA) and Interchange Format part-1 Introduction and General Principles, p, 38 (1989).
11) 伊藤，堀川，高野：仕様書エディタ (spec) の改良，第38回情報処理学会全国大会論文集，p. 1287 (1989).

(平成元年 7 月 12 日受付)
(平成元年 11 月 14 日採録)



堀川 博史（正会員）

昭和 30 年生．昭和 53 年名古屋工業大学情報工学科卒業．昭和 55 年北海道大学大学院工学研究科修士課程情報工学専攻修了．同年三菱電機(株)入社．同社情報電子研究所勤務．現在，ソフトウェア工学の研究開発を行っている．



伊藤 俊之（正会員）

昭和 35 年生．昭和 59 年名古屋大学工学部応用物理学科卒業．昭和 61 年同大学院工学研究科修士課程情報工学専攻修了．同年三菱電機(株)入社．同社情報電子研究所勤務．ソフトウェア工学，文書処理の研究開発に従事．

高野 彰（正会員）

昭和 23 年生．昭和 45 年学習院大学理学部数学科卒業．昭和 47 年同大学院修士課程修了．同年三菱電機(株)入社，情報電子研究所に勤務．ソフトウェア工学の研究開発に従事．平成元年より同社技術本部．電子情報通信学会会員．