

ドキュメントの構造化によるソフトウェア開発支援

辻 誠治 向井 雅樹 今井 良彦
松下電器産業株式会社 情報通信関西研究所

グループでの作業分担によるソフトウェア開発においては、作業者間での仕様変更の通知時間のずれや、インターフェース仕様の記載洩れが起るといった問題が生じる。これらの問題の解決のため、我々は、開発作業間で正確かつ迅速な情報の伝達が行なえる電子的なドキュメント作成環境の構築に取り組んだ。この環境では、ソフトウェア開発のための標準ドキュメントの構造を定義する機構が設けられており、開発標準に沿ったドキュメントの作成が容易である。また、ドキュメント構造の構成要素間の関係を明記する手段を設け、複数のドキュメント間の整合性を保つことにより情報伝達の環境が改善できた。

Supporting development of software based on document structuring

Seiji Tsuji Masaki Mukai Yoshihiko Imai
KANSAI INFORMATION & COMMUNICATIONS RESEARCH LABORATORY
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD
1006, KADOMA, KADOMA-SHI, OSAKA 571 JAPAN

When software is developed by a group, there arise problems as to time lags of notification about specification changes between developers in the group, lack of information in documents on interface specification, etc. We have prepared a mechanism to define structures of normalized documents used in software developments, so that we can easily create documents according to standard methods of the software development. We have also achieved a way to retain the integrity of documents by using a facility for defining relations between elements that compose a document. We have made it possible to transmit information between developers correctly and rapidly.

1 はじめに

ソフトウェアのグループ開発において重要となることとして作業員間の正確な情報伝達が挙げられる。小人数によるソフトウェア開発と大人数でのソフトウェア開発を比較すると、多くの場合、前者の方が1人当たりの生産性は高い。これは、一般にメンバー数が増加すると、それだけコミュニケーションパスが複雑になったり、長くなったりすることが知られている。記録の残るドキュメントにより情報伝達の正確さを確保することが課題であり、ドキュメントの生産性の改善が期待されている [1][2]。

実際のソフトウェア開発現場におけるソフトウェア開発の工程毎のドキュメント作成作業について調査した。同一の工程における作業員間でのドキュメントの品質(均質性)の保持は、ドキュメントの標準化によって行われていた。重要項目の記載洩れをなくし、見易さを確保するため、ドキュメント量の多い関数仕様書、テスト成績一覧などに、帳票が採用されている。しかし、ソフトウェア開発現場で扱われている標準ドキュメントをみると、関数仕様書など工程毎に、9割程度の項目が同一であり、1割程度がプロジェクト固有の項目があることがわかった。

分担作業を並行して進めるために、少なくとも作業員は、プログラミング、テストやドキュメント作成などの作業を終了し、他に影響を及ぼす時点では必ず情報伝達を行い、作業を進めている。特に、システムテストから運用、保守での工程においては、成果物に関係をもった情報伝達の頻度が多くなっていることがわかった。

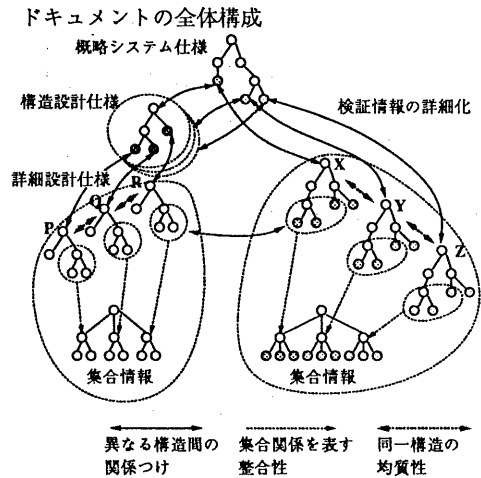
最近では、ネットワークを介し離散した環境での協同開発が可能になり、異なるシステム環境の間での情報交換が必要になってきている。そのため、デバイスや出力系に依存しない汎用的なドキュメント記法による情報伝達が重要である。

以上を整理すると、ソフトウェア開発におけるドキュメント処理に対する要望には次の2つの主要なことがあると言える。

- 標準ドキュメント体系を推進できる汎用的なドキュメント記法
- ドキュメントを含む作業間での作業完了などを表す情報伝達機能

2 ソフトウェア開発ドキュメントの特徴

ソフトウェア開発における正確な情報伝達のために、ソフトウェア開発において作成されるドキュメントを分析調査した結果、問題とされる原因は、以下のようなドキュメントの品質に関わる3種の要因に分類できた。



ドキュメント作成の作業経過例

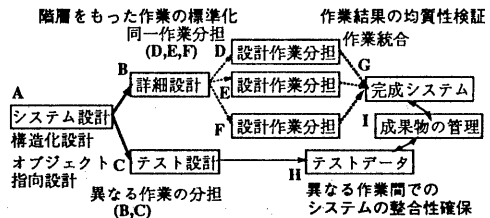


図1: グループでのソフトウェア開発におけるドキュメントの全体構成と作業経過例

1. 同一工程内でのドキュメントの均質性
グループでの同一作業を分担する時に起こる開発作業員間のバラツキをなくし、均質な開発情報の

品質を確保しなければならない(図1の同一作業分担D、E、F)。開発情報を均質にするために、分担作業を始める前に標準ドキュメント体系が作成され、同時に行われる作業の中で均質な情報を確保できるようにしている。図1のドキュメントP、Q、RやX、Y、Zの間では、構造が等しいというドキュメントの均質性を保っている。

しかし、設計の詳細化が進み、グループが細分化された時、共通する情報項目はあっても、グループ内で情報伝達すべき新たな項目が加わる。グループ間のインターフェースを考慮して標準ドキュメントを作成する場合、情報の記載洩れの防止と余分な情報作成の強要を防止しなければならない。そのため、グループ間で標準とすべき情報とグループ内だけで交換すべき情報を階層化し、階層構造のレベル分けによる選択機能が必要である。

詳細設計書などの場合、図2のようなA、B、C間に共通する情報項目7が、9割程度であり、2つのグループ間で共通する4、5、6とグループ内で独自である1、2、3の情報項目を、下段のように標準化レベルを設けて、情報伝達の対象となるグループによって伝達すべき項目を選択する必要がある。

グループ作業における情報伝達のレベル分けの例として、Object Lens がある [3]。

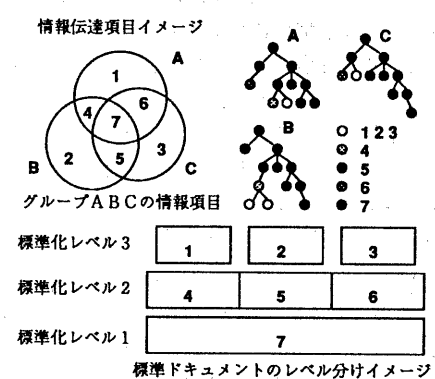


図2: 標準ドキュメントのレベル分け

2. 分担作業結果を表す集合ドキュメントの整合性

1つの作業を分担した場合、作業の終了後、分担した作業をまとめて1つの成果としなければならない(図1の作業統合D、E、F→G)。この時、分担して作成したドキュメントを集めて、作業結果全体を表すドキュメントを作成する。このような一覧表などの情報を集めて作成するドキュメントをここでは、集合ドキュメントと呼ぶ。集合ドキュメントは、また、分担開発作業中に作業経過や作成されたドキュメント間のバラツキなど品質を評価するために作成される。集合ドキュメントは分類や統計情報を表し、開発作業の変更がある度に繰り返し作成される。図1の(P,Q,R)、(X,Y,Z)と下記の集合情報が、作業分担して作成されるドキュメントと集合ドキュメントの関係を表している。

3. 構造の異なるドキュメント間の情報の整合性

製造や検証など目的とする作業が異なる場合、作成されるドキュメントの構造も異なる。異なる作業を分担する時、情報伝達内容の均質化の問題とは異なり、情報の通知洩れや遅延といった問題が重要である。この情報伝達のタイミングに関する問題は、UNIXにおけるmakeがソースコードの更新時に再コンパイルや再リンクを実行するのと同じように、ドキュメントの更新により影響のある作業を確実にを行う仕組みが必要である。しかし、一般にドキュメント作成は、プログラミングのように処理が定型化していないことが多く、作業者にドキュメントの更新通知を行い作業を進めることを、ここでの取り組みとした。

図1の成果物の管理のように階層設計、製造によるシステムの成果物とテスト作業の関係(リンク)を基に作業間で緊密な連絡を取りながら保守作業が進められている。

3 ドキュメント品質の改善方法

ソフトウェア開発における正確な情報伝達を達成するためにドキュメントの品質を確保する次の3点を解決方法のポイントとして挙げた。

1. ドキュメント構造を定義し、構造に沿ったドキュメント作成をする。

現在、レイアウト独立なドキュメントのもつ論理構造による構造化ドキュメントを表記するための標準化案として SGML が提案されている [4][5]。ドキュメント構造を定義するための文法が提案されており、SGML の利用者が宣言的にドキュメント構造を設定できる汎用性をもったものである。この考え方を基にソフトウェア開発に必要な標準ドキュメントの論理構造を定義できる仕様を規定した。また、SGML や TeX などと同じように識別子（識別子をタグと呼ぶ）を構造化のための情報として用いている。

このタグ情報に対して内容を記述することにより、開発情報の構造化ドキュメントを作成する。作成した構造化ドキュメントは、タグに対応した内容の判断を行う手続きを実行することができる。また、論理構造の階層化レベルを設定して情報伝達に必要な項目を選択することができる。図 3 の () 内の (親ノード:子ノード、子ノード) のような記述方法により、ドキュメントのもつ構造を表す。また、AAA、BBB、CCC のように構造化ドキュメントは記載される。

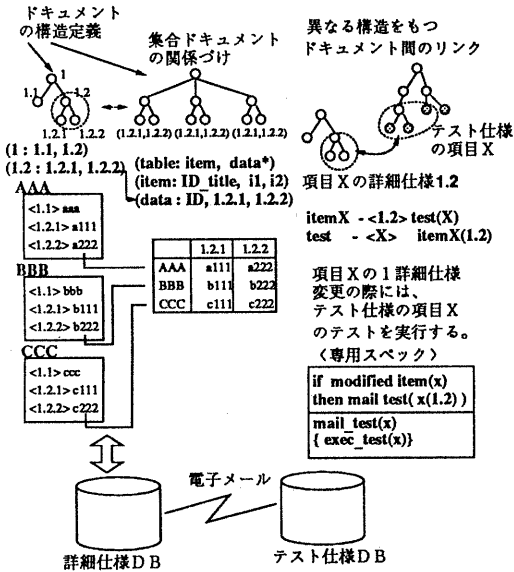


図 3: ドキュメント品質の改善イメージ

2. 開発情報と集合ドキュメント間の構造の関係を定義し、集合ドキュメントを作成する。

開発情報の論理構造とこの開発情報を集めた集合ドキュメントの論理構造間の関係を各々の論理構造を示すタグを基に定義する。この関係を表す情報をもとに複数の開発情報から awk などのテキスト処理ツールを用いて、集合ドキュメントを生成する機構を設ける。この機構により、開発情報の更新に伴い繰り返し作成する集合ドキュメントの作成効率を向上できる。図 3 の例では、AAA、BBB、CCC は、() 内の * により集合情報を表し、同一項目タグを指定して、関係づけることにより、図の中央の表形式のドキュメントを生成することを表している。

また、複数の開発情報を 1 つのデータ集合体として扱う場合、データベースを利用することが有効である。ドキュメント構造の項目 (タグ) をデータベースのスキーマに関係づけて、構造化ドキュメントをデータベースへ登録し、検索することができる [6]。データベースを用いることにより、高速で豊富な検索項目を設定することができる。

集合ドキュメントの出力形状として帳票は、画面や紙の2次元空間を有効に利用し、集合ドキュメントの項目毎の検証を容易にするため、構造化ドキュメントを帳票出力する機能が必要である。

- 異なる作業間で開発情報の変更通知を行う機構を設ける。

異なる作業間において、開発情報の変更通知により変更に対応した情報の自動変更は困難である。しかし、makeなどの機能と同じように更新を作業のきっかけとして、開発情報間で変更通知を電子メールなどにより、通知することが可能である。成果物としてドキュメントを詳細なレベルで情報を関係づけることにより、変更作業により通知洩れや遅延が解消できる。

このために構造化ドキュメントで表現された開発情報のタグに、直接または間接的にドキュメント間の関係を表す情報を明記する。図3の専用スペックのような記述を行い、ドキュメントの更新時に動作を起こす。今回作成したソフトウェア開発支援システムでは、データベースに変更通知の機構を持たせることにより、情報伝達を実現した。

4 ソフトウェア開発支援システムの構成

我々は、ドキュメントの品質を確保するために、構造化ドキュメントを流通の基盤とし、この構造化ドキュメントを活用するための基本ツール群を開発した。基本ツール群には、帳票作成ツール、DB__IFツールとデータ交換ツールがある。基本ツール群を利用して正確な情報伝達を支援するソフトウェア開発支援アプリケーションの開発を行った。今回開発したシステムの全体構成を図4に示す。

4.1 構造化ドキュメント

ドキュメントの論理構造を定義し、ドキュメントの構造を表すタグをもとに構造化ドキュメントを作成できる規則作りをした。

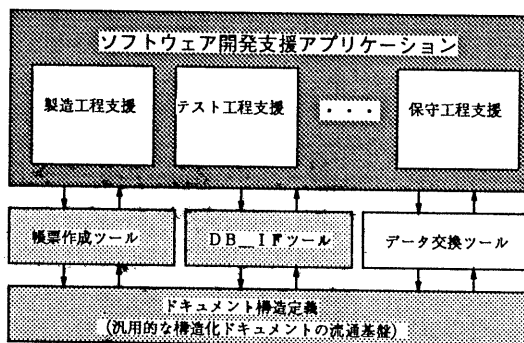


図4 ソフトウェア開発支援システムの構成

4.2 基本ツール群

- 構造化ドキュメントからソフトウェア開発の標準ドキュメントとして多数用いられている帳票を作成する帳票作成ツールを開発した。
- 構造化ドキュメントのDBへの登録、取り出しのためにSQLによる汎用インターフェースツールDB__IFツールを開発した[7]。
- ソフトウェア開発対象アプリケーション毎に、ドキュメントの構造間関係をもとにawkなどのテキスト処理ツールを利用して、構造化ドキュメントの構造を変換するデータ交換ツールを開発した。

4.3 ソフトウェア開発支援アプリケーション

構造化ドキュメントを利用して、ソフトウェア開発の製造、テスト、保守工程における作業支援を行うアプリケーションを開発した。

5 帳票作成ツール

ここでは、前章で述べた基本ツール群の内、帳票作成ツールについて説明する。帳票作成ツールは、UNIX上で動作する構造化ドキュメントの出力に適したデバイス独立なツールである。

5.1 構成

帳票作成ツールは、図5に示すように、定義ファイルとデータファイルを入力データとし、Xウィンドやプリンタに帳票を出力する。定義ファイルには、ドキュメントの論理構造、書式情報およびこれらを構成する項目毎の関係を表す情報が記述されている。データファイルは、定義ファイルで定義された論理構造に従った構造化ドキュメントである。

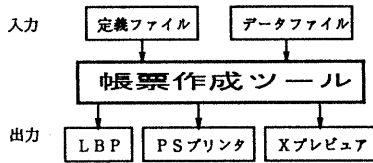


図5: 帳票作成ツールによる処理の概略

5.2 機能

ドキュメントが構造化されていることを利用し、データファイルの内容を変えずに帳票出力の形状を変更ができる。基本機能として次のようなものがある。

1. 標準帳票枠の設定と表形式のマクロを組合わせて、自由な項目設定が可能な表形式の帳票ドキュメントが作成できる。
2. 仮想的な2次元領域にデータを表記する機能を持ち、二段組などの自由な形状の変更の他、差し込み印刷のイメージで出力できる。
3. 文字フォントや文字ピッチなどの属性情報をデバイス毎に設定できる。また、データファイルの書式位置へのテキスト量が多い場合に文字間の調整などを設定できる機能をもつ。

5.3 例

データファイルに記述されているタグ付きの構造化ドキュメントから、定義ファイルの書式内容によって帳票出力形状を選択できる例を図6に様子を示す。ここでは、一つの関数仕様書に関するデータファイルから、グ

ループの階層によって提出すべき帳票の出力選択ができることや、納品先の帳票への出力選択ができることを示している。

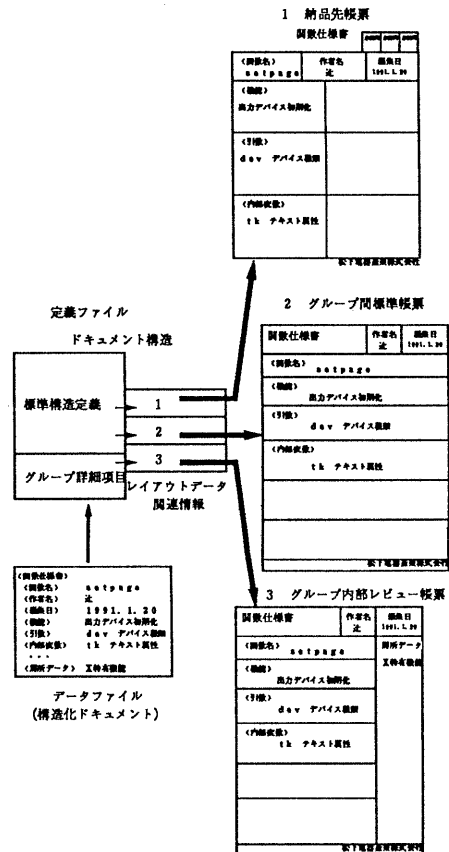


図6: 帳票作成ツールの活用例

6 ソフトウェア開発現場への適用

ソフトウェア開発現場において帳票作成ツール、DB __IF ツールおよびデータ交換ツールを用いて有効活用する例を示す。

6.1 テスト工程への応用

ソフトウェアの品質の確保のためテストは重要である。テスト作業の特徴として成果物の改変に従い、テストが繰り返し実行されることが挙げられる。また、システムテストなどのブラックボックステストでは、完全な品質のためのテスト項目を網羅することはできない。そのため経験的な手段により、テスト仕様を変更することが多い。このような現状に即したテスト支援のため、我々は、構造化ドキュメントとしてテスト仕様書、成績書（結果）を定義し、定義したこれらのテストに関する情報を容易に登録・検索するためデータベースを利用したテスト支援システムを開発した。基本ツール群の活用内容と、テスト支援システムのために新規開発した内容について説明し、このテスト支援システムで得られた特長について記す。

基本ツール群の活用部分

- テスト仕様データのデータベースへの登録および検索の機能
- テスト結果のデータベースへの登録機能
- テスト仕様書、成績書の出力機能

テスト支援システム用の新規開発部分

- テスト仕様からテスト実行データへの変換機能
- テスト結果のテスト成績書（結果）データへの変換機能

テスト支援システムの特長

- バッチテストの自動化と自動成績書作成
- 対話テストの項目整理および成績書記表枠の作成

テスト仕様に従ったテスト作業の正確な繰り返しと、テスト作業とテスト結果を示すドキュメントの一貫性を保ち、テスト作業効率の改善が達成できた。

6.2 保守工程への応用

ソフトウェア開発によってできたシステムの保守現場においてシステムの品質を維持向上させるためには、利用者から報告された障害に対する対応を迅速に行うことが不可欠である。障害対策は障害の発見者、障害の管

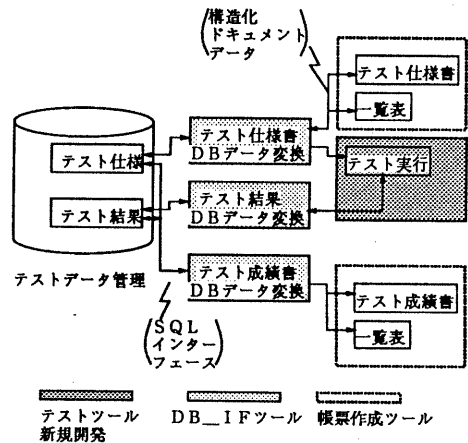


図7: テスト工程への活用例

理者、開発担当者および検証者の間で情報交換を行いながら作業を進めるものである。従来よりソフトウェア保守の現場では帳票を用いた情報交換の作業が行われてきた。同様の作業を電子メールとDBを用いた障害管理システムを利用して行うことにより保守工程における作業が改善できた [8]。

障害管理システムの利用に際し、障害の発見者、障害の管理者、開発担当者および検証者の間で行われる電子メールによる情報交換に構造化ドキュメントを用いた。これにより発見者が直接DBにアクセスするのではなく、電子メールで送信された障害情報を障害の管理者がDB-IFツールを用いてDBに入力することが可能になった。発見者の負担を軽減し、障害情報を直接アクセスできる人を限定できた。

障害管理システムにより、障害状況を必要に応じた情報を検索し、異なる形状のドキュメントを作成することが可能になった。また、システムの稼働状況や障害発生状況を表すドキュメントの作成が容易になった。

障害管理システムによるシステム保守の様子を図8に示す。

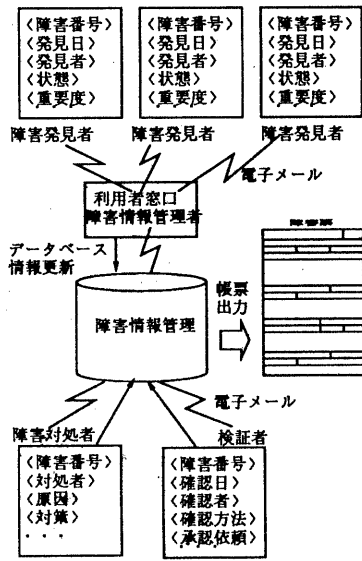


図 8: 保守工程への応用例

7 まとめ

構造化ドキュメントを扱う帳票作成ツール、DB __IF ツールなどの基本ツール群を用いることによって、ソフトウェア開発現場における情報伝達の基盤作りができた。標準ドキュメントの構造を宣言的に定義することにより、必要とされる標準化のための階層化と選択が実現できた。また、階層構造を示すタグ情報を基に内容に記載洩れや正確さを判断する機構を設けることができた。DBの利用により、ドキュメントデータの一貫性管理や高速で組合せが豊富な情報検索ができた。これを用いて帳票作成など集合ドキュメントの出力により、ドキュメントデータの正当性などの評価が容易になった。また、電子メールなどを利用して情報伝達する際、論理構造に従った構造化ドキュメントの交換により、デバイス独立が実現できた。加えて、ドキュメント作業と情報伝達の間の関係づけにより迅速で通知洩れのない環境を実現できた。さらに、この他、直接的な効果ではないが、ソフトウェア開発作業の履歴を電子状態ととることができるため、開発作業を定量化する基盤になるものと期待して

いる。

今後、ソフトウェア開発支援を進める上で課題となっているソフトウェア開発作業の作業分析と計量化に、今回開発したこのソフトウェア開発環境を活用していく予定である。また、構造化ドキュメント間の関係づけ作業の汎用化を始め、カスタマイズ機能の充実を図り、ソフトウェア開発支援システム自身の生産性を上げていきたい。

参考文献

- [1] 花田:ソフトウェアの仕様化と設計, 日科技連ソフトウェア品質管理シリーズ
- [2] (財) 日本規格協会:平成 2 年度ソフトウェア開発・システムの文書化標準化調査研究報告書, (財) 日本規格協会
- [3] Kum-Yew Lai, Thomas W. Malone, and Keh-Chaing Yu: Object Lens: A "Spreadsheet" for Cooperative Work. ACM Transactions on Office Information Systems, Vol.6, No.4, October 1988
- [4] International Standard ISO 8879-1986(E) Information processing - Text and office systems - Standard Generalized Markup Language (SGML)
- [5] Martin Brian 著, 山崎 訳: SGML 入門, 株式会社アスキー
- [6] 芝野: SGML と全文データベース, 情報学基礎研究会 89-FI-14-2, Vol.89.No66,p1-8(1989)
- [7] 谷川他:SGML Document_DB および伝達制御機能を持つドキュメント管理システムのプラットフォーム, 日本ソフトウェア科学会 (関西)SW-91-8-5(1991.3.22)
- [8] 谷川他:グループ開発を支援する障害管理システム, 平成 2 年電気関係学会関西支部連合大会 S8-9