

7N-4

ソフトウェア設計ドキュメントの入力の標準化に関する一考察

藤井 義信¹ 清兼 幸雄¹ 宮川 純一¹ 藤井 孝夫² 原口 哲治² 松田 浩伸²¹富士通株式会社 ²富士通関西通信システム株式会社

1. はじめに

近年ソフトウェアの開発は大規模化かつ短サイクル化している。ソフトウェア面では各種ツールやエンジニアリング手法を使用しての開発効率化が行われているが、膨大の量のドキュメントに関しては有効な開発、活用がされておらず、文書構造、文書内容、記述レベルの統一はできていないことが多い。

本稿では、ソフトウェア開発作業において作成される各種設計ドキュメントに文書記述の為に国際標準規約であるSGMLを適用し、文書の標準化を行う方法について検討した。

2. 設計ドキュメントの標準化アプローチ

2.1 ソフトウェア設計ドキュメント入力の現状

ソフトウェア開発においては、プログラム開発を急ぐあまり設計ドキュメント作成に費やす時間が削減される傾向がある。このような状況で作成されたソフトウェアは結果的に手戻りやバグが多く、開発作業は非効率になっている。ソフトウェアの仕様をきちんと定義し、開発作業の効率化を図る為に、まずは論理的に記述され、内容の完備した設計ドキュメントを作成することが重要である。

2.2 SGMLの適用

ドキュメントの標準化には、構造の標準化と内容の標準化の2つが必要である。ソフトウェア開発では、設計ドキュメント用の執筆規約が存在しないという場合もあるが、執筆規約は存在するがうまく参照できていない為に規約が守られていないことも少

なくない。そこで、標準化されたドキュメント（構造、内容）を明確に記述する言語としてSGML（Standard Generalized Markup Language）を導入することで、標準化の徹底と共有情報の活用を図ることができる。

SGMLでは論理構造と体裁情報の分離が行われており、DTDに定義されている論理構造だけを意識すれば良いようになっている。タグと呼ばれるマークによって構造を記述するというSGMLのルールに従って記述すれば、文書の標準化が徹底され、再利用が容易になる。

3. 設計ドキュメント入力試行

3.1 試行環境（前提条件）

設計ドキュメントの標準化にSGMLを適用した場合の効果を評価する為に、設計ドキュメントの入力試行を行なった。本試行においては、マルチウィンドウOS上で、SGML専用エディタと図形エディタを利用し、基本検討/設計書の入力を行い、その使い勝手を評価した。（図1）

3.2 ドキュメントの入力条件

SGMLエディタの多くは、DTDにおいて定義された文書構造を自動的に解析し、構造的に不正な入力を制限することが可能である。但し、入力にはある程度SGMLのタグと構造を意識することが必要になる。設計者にはできるだけ余計な負担をかけることなく、あらかじめ文書の雛形が入力されているテンプレート文書を用意し、その文書を元に更新

A study of standardization of software design documents

Yoshinobu FUJII¹, Yokio KIYOKANE¹, Junichi MIYAGAWA¹, Takao FUJII², Tetsuharu HARAGUTI², Hironobu MATSUDA²

¹FUJITSU LIMITED, ²FUJITSU KANSAI COMMUNICATION SYSTEMS LIMITED

するようにした。また、入力時には必要でない限りタグを隠した状態のままを入力することにより、SGMLを意識させないようにした。

また、内容の標準化の為に、設計規約や記述例を表示するガイダンス機能が必要となる。今回の試行ではできる限り既存機能を利用することにし、ドキュメント中に設計規約や記述例をエンティティとして埋め込み、表示/非表示を切り替えることで実現した。

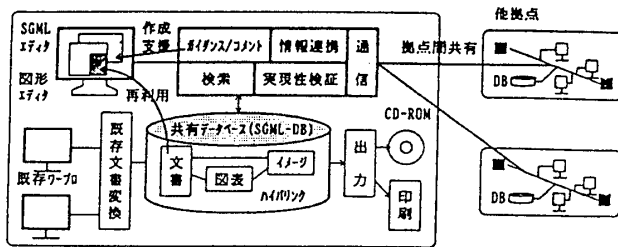


図1 SGMLを適用した設計ドキュメント入力環境

4. 試行結果と考察

4.1 問題点

上記入力試行を行なった結果、以下の問題点が顕在化した。

- ・SGML専用エディタを使用すると、タグとDTDを全く意識しない文書入力是不可能である。可能な限りタグを意識させない入力システムの方が良い。
- ・入力途中での文書構造の変更が困難かつ手間がかかる。規約を守りつつ容易に構造の変更ができる入力システムが必要である。
- ・図形とテキストは混在した形で表示される必要がある。また、編集も同一ウィンドウ内でできた方が良い。
- ・必要な部分だけのガイダンスを必要な時に瞬時に参照することができない。
- ・ドキュメント構造が不安定な時点で複雑なDTDを定義すると、構造が変更される毎にDTDや既存文書に変更が必要になる。

その他に、入力したいタグをどの位置で挿入できるかを表示する機能の不足、入力画面と印刷イメージの不一致、レスポンスの遅さ、操作の複雑さ、等が問題として挙げられた。これらの問題は、SGML

専用エディタがSGMLの機能を活用することを優先させ、ユーザ要求やユーザインタフェースの考慮が不十分である為と思われる。

4.2 必要条件

上記問題を解決する為には、ユーザが必要としている機能を考慮し、必要とされている機能が複雑な場合でも簡単なユーザインタフェースを提供する設計ドキュメント入力システムを構築し、使い勝手を向上させることが必要である。また、設計書用のDTDはドキュメント構造の変更に柔軟であることが望ましい。(表1)

表1 設計ドキュメント入力環境の必要条件

項目	必要条件
設計書エディタ (SGMLエディタ + 図形エディタ)	入力中での構造変更が容易
	タグを意識せずに構造的記述が可能
	図形とテキストが混在可能
入力支援ガイダンス	必要なガイダンスだけが瞬時に参照可能
文書型定義 (DTD)	変更に対する柔軟性

5. おわりに

本稿では、ソフトウェア開発作業において作成される各種設計ドキュメントの標準化の為にSGMLを適用した場合の設計ドキュメント入力環境について考察した。しかし、試行結果から導いた設計ドキュメント入力環境としての必要条件を全て満たす文書入力システムは現時点では存在しないと思われる。今後は、既存の入力システムをカスタマイズする形で、設計ドキュメント作成に最適な文書入力システムを構築することを目標に考察していきたい。

参考文献

- 1) 原口, 他: ソフトウェア開発におけるSGMLを用いた情報共有化方式の一考察, 情報処理学会第45回全国大会, 1992年
- 2) 宮川, 他: ソフトウェア設計情報の共有化方式の一考察, 電子情報通信学会交換システム研究会, 1993年
- 3) 阪口, 他: 設計情報共有化へのSGMLの応用, 情報処理学会第48回全国大会, 1994年
- 4) 吉岡, 他: SGMLのススメ, オーム社, 1993年